UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE BIOLOGIA ANIMAL

Classificação taxonómica de procariotas com base em sequências simuladas do gene *16S rRNA*

Luís Miguel Ramos Vieira

Mestrado em Bioinformática e Biologia Computacional
Especialização em Bioinformática

Dissertação orientada por:
Prof. Octávio S. Paulo

2017

Agradecimentos

Agradeço à Célia Ventura pelo apoio na análise dos resultados do *mothur* e do *QIIME*, e à Catarina Silva pela revisão crítica do texto.

Resumo

O estudo de fragmentos de DNA obtidos directamente de uma amostra ambiental é designado por metagenómica. A determinação da sequência de bases desses fragmentos pode ser obtida através da sequenciação de todos os fragmentos da amostra (sequenciação *shotgun*) ou de amplicões de genes marcadores, como por exemplo o gene *16S rRNA*. Nos últimos anos, os estudos de metagenómica têm tido um desenvolvimento crescente em resultado da introdução de novas plataformas de sequenciação paralela massiva, que permitem obter várias centenas de gigabases de sequência por ensaio. Apesar do potencial de conhecimento científico que estes estudos vieram permitir, colocaram também novos desafios na análise do grande volume de dados obtido. Assim, a necessidade de análise de dados de sequenciação *shotgun* ou de amplicões do gene *16S rRNA* despoletou o aparecimento de múltiplas ferramentas bioinformáticas que cobrem os diferentes níveis de análise de metagenomas, desde a avaliação da qualidade das leituras de sequenciação até à identificação de novos genes com relevância funcional. No presente trabalho reviram-se mais de uma centena de programas disponíveis no domínio público que podem ser aplicados à análise de dados de sequenciação de metagenomas, incluindo 91 programas que permitem a identificação taxonómica das leituras obtidas na sequenciação. No entanto, é um facto que programas distintos, aplicados ao mesmo conjunto de dados, podem produzir resultados diferentes. De forma a testar e comparar a performance dos programas de classificação taxonómica de leituras do gene *16S rRNA*, foi desenvolvido um programa (*sim16S*) em linguagem *Matlab* que permite obter leituras simuladas de amplicões deste gene, escolhidos a partir de uma base de dados de sequências de referência usando oligonucleótidos introduzidos pelo utilizador. O *sim16S* produz outros ficheiros de dados, incluindo o número de leituras atribuídas a cada táxon dos 5 níveis taxonómicos desde o filo até ao género, e um relatório com diversas estatísticas. Neste trabalho, o *sim16S* foi utilizado para produzir diversos conjuntos de leituras de 2 amplicões do gene *16S rRNA* e introduzir substituições de bases, de acordo com um modelo estatístico que simula a distribuição de erros de sequenciação. Com base nestes conjuntos de leituras, foram efectuadas 20 análises de classificação taxonómica em paralelo com os programas *QIIME* e *mothur*, que constituem os 2 programas mais citados neste âmbito na literatura científica. A análise de leituras sem erros de sequenciação mostrou que a exactidão da classificação taxonómica decresce em direcção aos níveis taxonómicos inferiores, mesmo utilizando as sequências que deram origem às leituras simuladas como base de dados de referência. A utilização de outras bases de dados nos 2 programas conduziu a um aumento significativo de táxones sem classificação taxonómica completa, em todos os níveis taxonómicos. A presença de 1, 2 ou 4 erros de sequenciação nas leituras não afectou a classificação taxonómica das leituras nos níveis de filo, classe e ordem em ambos os programas, relativamente à classificação das leituras sem erros. No entanto, a exactidão da classificação no *mothur*, nos restantes níveis taxonómicos, foi afectada na presença de ~1%, ~10% e 100% de leituras com 1 erro de sequenciação por leitura ou ~10% de leituras com 2 ou 4 erros por leitura. Pelo contrário, o *QIIME* apenas revelou uma exactidão inferior a 99% nos conjuntos de leituras com 100% de leituras com 1 erro, sugerindo que este programa é menos sensível à presença de erros de sequenciação do que o *mothur*. As análises efectuadas mostraram que o *sim16S* é uma ferramenta bioinformática útil para testar a performance da classificação taxonómica de diferentes programas existentes no domínio público. Além disso, o *sim16S* pode facilmente ser adaptado a outros genes procariotas ou eucariotas para os quais estejam disponíveis bases de dados de sequências de referência, podendo assim funcionar como uma ferramenta de âmbito geral no contexto dos estudos de metagenómica.

Palavras-chave: metagenómica, gene *16S rRNA*, sequenciação de amplicões, simulação de leituras.

Abstract

The study of DNA fragments obtained directly from an environmental sample is called metagenomics. Determination of the sequence of bases of these fragments can be achieved by sequencing all fragments in the sample (*shotgun* sequencing) or amplicons derived from marker genes, such as the *16S rRNA* gene. In recent years, metagenomics studies have been growing as a result of the introduction of new massive parallel sequencing platforms, which allow for several hundred gigabases of sequence per assay. Despite the potential of scientific knowledge that these studies allowed, they also posed new difficulties in the analysis of the large volume of data obtained. Thus, the need for analysis of *shotgun* sequencing or *16S rRNA* gene amplicons triggered the emergence of multiple bioinformatics tools covering the different levels of metagenome analysis, ranging from the quality evaluation of sequencing reads to the identification of new genes with functional relevance. In the present work, more than 100 publicly available programs that can be applied to the analysis of metagenome sequencing data were analyzed, including 91 programs that allow taxonomic identification of sequencing reads. However, it is a fact that distinct programs, applied to the same set of data, can produce different results. In order to test and compare the performance of the *16S rRNA* gene taxonomic classification tools, a program (*sim16S*) was developed in *Matlab* language that allows obtaining simulated reads of gene amplicons, chosen from a database of sequences using oligonucleotides introduced by the user. *sim16S* produces several data files, including the number of reads assigned to each taxon from the 5 taxonomic levels from phylum to genus, and a report with various statistics. In this work, *sim16S* was used to produce several sets of reads of 2 amplicons of the *16S rRNA* gene, in which base substitutions were introduced according to a statistical model that simulates the distribution of sequencing errors. Based on *sim16S* datasets, 20 taxonomic classification analyzes were carried out in parallel with *QIIME* and *mothur*, which constitute the 2 most cited programs in the scientific literature in this field. Analysis of reads without sequencing errors showed that the accuracy of the taxonomic classification decreases toward the lower taxonomic levels, even using the sequences that gave rise to the simulated reads as a reference sequence database. The use of other databases in the two programs led to a significant increase in incomplete classified taxa at all taxonomic levels. The presence of 1, 2 or 4 sequencing errors in the reads did not affect the taxonomic classification at the phylum, class and order levels in both programs, relative to the classification of error-free reads. However, the accuracy of *mothur* classification at the remaining taxonomic levels was affected in the presence of ~1%, ~10% and 100% of reads with 1 sequencing error per read or in the presence of ~10% of reads with 2 or 4 errors per read. In contrast, *QIIME* only showed an accuracy of less than 99% in read sets with 100% of reads with 1 error, suggesting that this program is less sensitive to the presence of sequencing errors than *mothur*. These studies showed that *sim16S* is a useful bioinformatics tool to test the accuracy of the taxonomic classification of different programs available in the public domain. In addition, *sim16S* can easily be adapted to other prokaryotic or eukaryotic genes for which sequence databases are available and can thus function as a general tool in the context of metagenomics studies.

Keywords: metagenomics, *16S rRNA* gene, amplicon sequencing, read simulation.

Índice

Lista de figuras

Lista de tabelas

1. Introdução

A **metagenómica** designa o estudo dos fragmentos de DNA que podem ser obtidos directamente de uma amostra ambiental. No caso da metagenómica, o termo ambiental não se refere somente a um ambiente natural como o solo ou a água de um rio, mas pode também incluir as superfícies externas ou as cavidades internas do corpo humano ou de outros animais. O termo foi cunhado por Jo Handelsman e colaboradores em 1998, no contexto da hipótese de clonagem dos genomas existentes na microflora do solo (Handelsman *et al.*, 1998). Há cerca de 12 anos atrás, uma equipa de investigadores da Universidade da Califórnia deu os primeiros passos nos estudos de sequenciação metagenómica, ao utilizar a tecnologia de sequenciação capilar para obter a sequência parcial ou quase completa, de 5 genomas de espécies distintas presentes num biofilme acidofílico natural (Tyson *et al.*, 2004). Desde então, a metagenómica tem adquirido uma relevância crescente na literatura científica, nomeadamente em consequência da evolução observada nas tecnologias de sequenciação de DNA. O papel relevante da sequenciação de DNA resulta do facto de a vasta maioria dos microorganismos não crescer em condições de cultura laboratorial (Hugenholtz *et al.*, 1998), pelo que apenas a análise das sequências de DNA, extraídas directamente da amostra ambiental, poderá fornecer informação biológica relevante para a maior parte dos organismos. Esta vantagem constitui o aspecto mais diferenciador entre a sequenciação genómica e a sequenciação metagenómica, uma vez que no primeiro caso é necessário proceder ao isolamento dos microorganismos e ao crescimento destes em condições de cultura laboratorial, antes de se efectuar a sequenciação de DNA (**figura 1.1**). Dada a grande diversidade de nichos ambientais onde podem ser encontradas comunidades de microorganismos, desde a superfície dos oceanos (Rusch *et al.*, 2007) até às várias partes do corpo humano (Nelson *et al*., 2010), os estudos de metagenómica têm vindo a encontrar cada vez mais campos de aplicação, nomeadamente na medicina humana (Relman, 2015). Ao conjunto de genomas presentes num dado nicho ambiental, como por exemplo a parte distal do intestino do corpo humano, atribui-se a designação de **microbioma** (Bäckhed *et al.*, 2005).

Até ao surgimento das tecnologias de "sequenciação de nova geração" (*next-generation sequencing*), que vieram possibilitar a sequenciação de genomas completos, os estudos de metagenómica baseavam-se essencialmente na sequenciação de genes "marcadores" cuja sequência apresenta elevada variabilidade nucleotídica entre diferentes grupos taxonómicos, mas que se mantém essencialmente constante dentro de uma mesma espécie. O reconhecimento das posições nucleotídicas variantes pode ser usado para permitir a identificação dos principais grupos taxonómicos presentes numa amostra ambiental. O exemplo mais conhecido é o do **gene *16S rRNA*** (também referido como *16S rDNA*), um gene conservado evolutivamente (Woese *et al.*, 1975), que codifica para a subunidade pequena do ribossoma das células procariotas, que fazem parte dos domínios *Bacteria* e *Archaea*. O potencial do gene *16S rRNA* na classificação de espécies procariotas, e a sua importância nos estudos de filogenética, são desde há muito tempo reconhecidos (Fox *et al.*, 1977). O gene tem aproximadamente 1500 nucleótidos de comprimento e é altamente constante nas estirpes já sequenciadas, podendo ser dividido estruturalmente em 9 regiões hipervariáveis, cuja sequência apresenta elevada variação nucleotídica entre os diferentes grupos taxonómicos. O número de genes *16S rRNA* pode variar entre 1 e 15 cópias em diferentes espécies, podendo apresentar variabilidade nucleotídica entre cópias até 1,23% num mesmo genoma, como é o caso da bactéria *Escherichia coli* (Klappenbach *et al.*, 2001). Devido à elevada diversidade nucleotídica do gene *16S rRNA* e à evolução tecnológica das plataformas de sequenciação, este gene é hoje em dia muito usado para caracterizar a estrutura das populações de microorganismos em amostras ambientais.

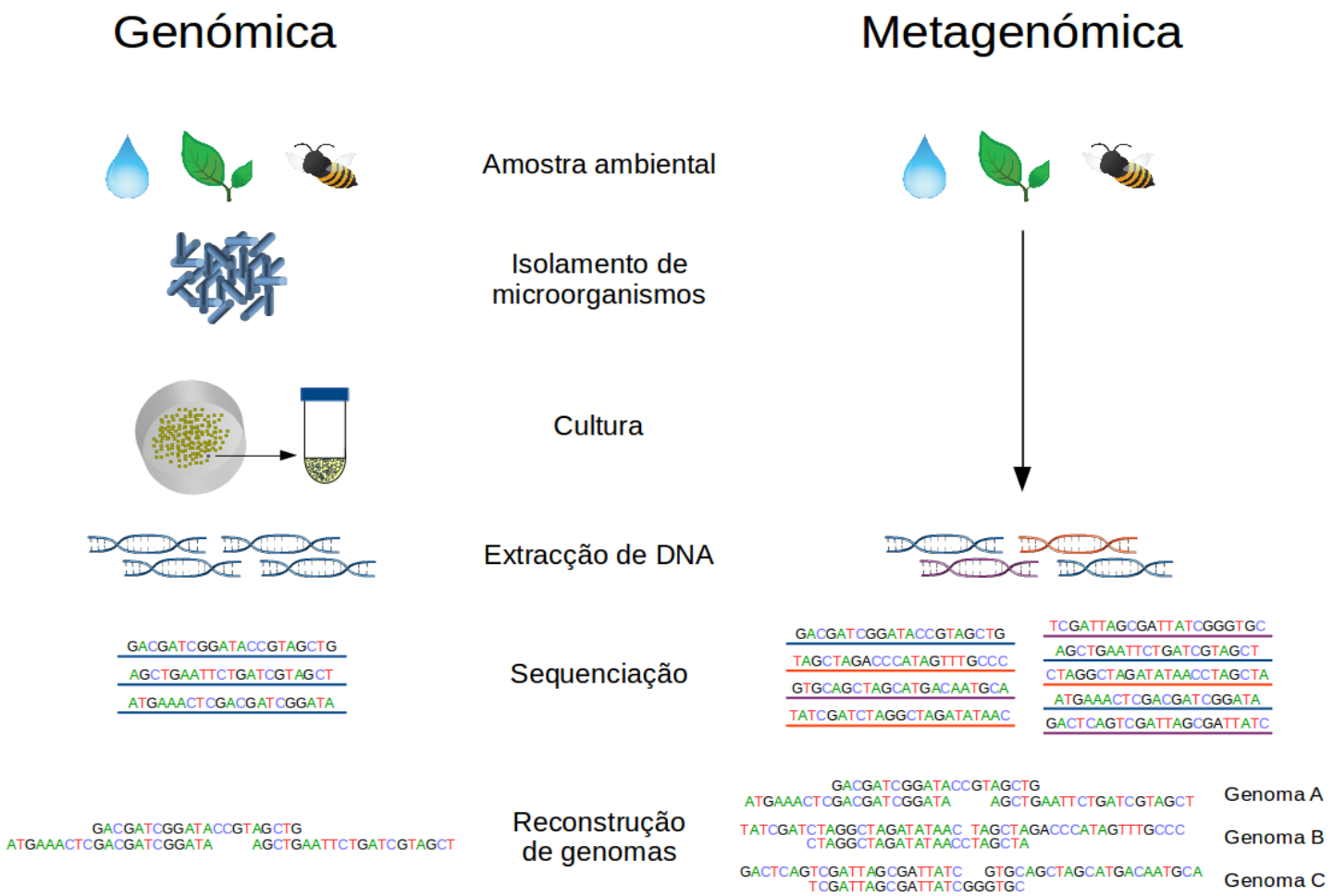

Figura 1.1. Esquema ilustrativo e simplificado das diferenças principais entre a sequenciação genómica e a sequenciação metagenómica. Na metagenómica, os genomas são sequenciados directamente a partir da amostra ambiental, sem necessidade de isolamento e/ou cultura dos microorganismos como no caso da genómica. Uma vez que os fragmentos de DNA dos vários genomas são sequenciados em paralelo, a maior dificuldade da metagenómica consiste em reconstruir cada um dos genomas presentes na amostra a partir de sequências curtas de DNA.

1.1. Tecnologias de sequenciação de nova geração

Desde 2005, ano em que foi lançado o primeiro sequenciador de "nova geração" da empresa 454 (mais tarde adquirida pela *Roche*), que utilizava a metodologia de pirosequenciação, vários fabricantes têm lançado plataformas de sequenciação massiva capazes de produzir, em cada ensaio, desde alguns megabases (Mb) de DNA (por exemplo, *Ion Torrent* ou *Roche 454*) até 1 ou mais terabases de DNA (no caso das plataformas *Illumina HiSeq* ou *NovaSeq*). O potencial das novas plataformas de sequenciação abriu assim oportunidades, até então inexistentes, para a sequenciação de comunidades de organismos com grande profundidade. Com a excepção das chamadas plataformas de sequenciação de "terceira geração", como as da *Pacific Biosciences* ou da *Oxford Nanopore Technologies*, que não requerem amplificação dos fragmentos de DNA pré-sequenciação, e permitem sequenciar moléculas de DNA de grande comprimento, a generalidade das plataformas actuais requer que a amostra de DNA seja inicialmente cortada em fragmentos de 100-1000 pares de bases (pb) de comprimento, através de técnicas como a sonicação ou digestão enzimática (por exemplo, através de transposases), ou amplificada através da técnica de *polymerase chain reaction* (PCR) usando oligonucleótidos específicos para a (s) região (ões) genómica (s) de interesse. Os fragmentos assim produzidos são ligados a sequências adaptadoras, específicas de cada tipo de plataforma de sequenciação, que permitem a ligação dos oligonucleótidos utilizados para a sequenciação. Além disso, uma das inovações metodológicas que surgiu com as novas plataformas de sequenciação, foi a possibilidade de sequenciar várias amostras em simultâneo num mesmo suporte sólido (por exemplo, *flow cell* ou *chip*). Neste caso, as amostras são previamente identificadas através de uma sequência única (normalmente com 6-8 nucleótidos de comprimento), designada índice (ou identificador molecular), que está inserida nas sequências adaptadoras de cada amostra. As bibliotecas construídas com índices distintos podem então ser misturadas para formar uma biblioteca única (*pool*), cujos fragmentos são sequenciados em simultâneo. A este processo dá-se a designação de *multiplexing*. A sequenciação destes fragmentos dá origem às *reads* (traduzido daqui em diante como "leituras"), que consistem numa sequência contínua de bases do DNA, a qual tem um comprimento fixo no caso das plataformas que produzem leituras curtas. No final da sequenciação, as leituras sofrem um processo de *demultiplexing* de acordo com os resultados da leitura dos respectivos índices, sendo adicionadas a ficheiros (por exemplo, *.fastq*) individuais para cada amostra (ver secção 1.3). Estes ficheiros são o ponto de partida para as várias etapas de análise de dados de genómica ou metagenómica.

1.2. Principais abordagens de sequenciação em metagenómica

Os desenvolvimentos tecnológicos e metodológicos atrás referidos vieram potenciar as aplicações das novas plataformas de sequenciação não só na área da metagenómica, mas também da metatranscriptómica (Gilbert *et al.*, 2008), a qual permite quantificar o perfil metabólico de uma comunidade de organismos num momento específico. No caso da metagenómica, podem actualmente ser utilizadas duas abordagens de sequenciação distintas. A metodologia mais comum consiste numa abordagem de **sequenciação dirigida** (do Inglês *targeted sequencing*), através da qual são amplificados por PCR um ou mais genes marcadores, que demonstrem elevada variabilidade de sequência nos grupos de organismos em estudo. No caso dos organismos procariotas, a sequenciação dirigida consiste maioritariamente na amplificação de uma ou mais regiões hipervariáveis do gene *16S rRNA*, utilizando oligonucleótidos específicos (Sogin *et al.*, 2006). O tamanho dos fragmentos da PCR deve ser adequado ao comprimento das leituras da plataforma de sequenciação, para que possam ser

sequenciados na totalidade evitando perda de informação biológica. As leituras dos fragmentos sequenciados são depois analisadas utilizando algoritmos informáticos e bases de dados específicas de sequências do gene *16S rRNA* (ver secção 1.7.1), que permitirão identificar o grupo taxonómico a que pertence cada fragmento de DNA. Não existe uma região hipervariável única que seja diferenciadora para todos os grupos de microorganismos, mas existem regiões que têm maior poder discriminatório do que outras para determinados grupos taxonómicos (Klindworth *et al.*, 2013). Assim, a sequenciação do gene *16S rRNA* pode implicar a amplificação de 2 ou mais regiões para se obter o poder de discriminação necessário para as comunidades de microorganismos em estudo.

Uma outra abordagem metodológica em metagenómica designa-se comummente por **sequenciação *shotgun*** e consiste em fragmentar as moléculas de DNA em cadeias de pequeno comprimento, e sequenciá-las sem qualquer tipo de selecção. No entanto, a análise bioinformática dos dados gerados pela sequenciação *shotgun* é muito mais complexa do que a utilizada na sequenciação dirigida. Uma vez que o genoma de cada microorganismo é sequenciado a partir de múltiplos fragmentos distintos, é necessário aplicar algoritmos informáticos específicos de forma a obterem-se *contigs* (i.e., conjuntos de leituras parcialmente sobrepostas entre si que formam uma sequência de DNA maior e ininterrupta), que permitam reconstruir a sequência genómica de cada espécie ou estirpe presente na amostra. Alternativamente, as leituras dos vários genomas podem ser classificadas taxonomicamente de acordo com a sequência de bases ou com características intrínsecas da própria sequência.

1.3. Controlo de qualidade e pré-processamento de leituras

As sequências geradas pelo equipamento de sequenciação devem ser submetidas a um processo de controlo de qualidade e de pré-processamento das leituras antes de poderem ser usadas nas etapas subsequentes de análise. Esta fase é habitualmente designada como "análise primária" e pode decompor-se nas etapas representadas na **figura 1.2**. A **tabela 1.1** apresenta uma descrição das funcionalidades principais de alguns programas que permitem realizar uma ou mais etapas iniciais da análise primária. Estes programas, assim como os descritos nas secções seguintes, resultaram de uma pesquisa da literatura biomédica existente na *PubMed* (National Center for Biotechnology Information - Pubmed, 1946) e da consulta da base de metadados *OMICTools* (Henry *et al.*, 2014). A maioria dos programas funciona em ambiente *Linux*/*Unix*, embora alguns possam também correr em *Windows* ou *MacOS*. As linguagens de programação mais utilizadas incluem o C, C++, *Python*, *Perl* e *Java*.

1.3.1. Avaliação de qualidade das leituras de sequenciação

A análise primária deve iniciar-se com uma avaliação global dos dados obtidos, incluindo a qualidade das bases sequenciadas, a proporção de cada base ao longo do comprimento da leitura, a percentagem do conteúdo guanina-citosina (GC) e a presença de sequências de adaptadores, usando por exemplo o *DeconSeq* (Schmieder and Edwards, 2011a), o *PRINSEQ* (Schmieder and Edwards, 2011b) ou o *FastQC* (Andrews, 2010). Além dos ficheiros de dados brutos (*.fastq*), alguns programas como o *FastQC*, o *htseq-qa* (Anders *et al.,* 2015) e o *SAMstat* (Lassmann *et al.*, 2011), permitem a análise de ficheiros de formato *.bam* e/ou *.sam*, que são produzidos após mapeamento das leituras de sequências no(s) genoma(s) de referência. Assim, é possível comparar as métricas de qualidade das leituras não processadas com as das leituras mapeadas, o que pode confirmar que o mapeamento dos genomas sequenciados foi efectuado usando maioritariamente leituras de boa qualidade.

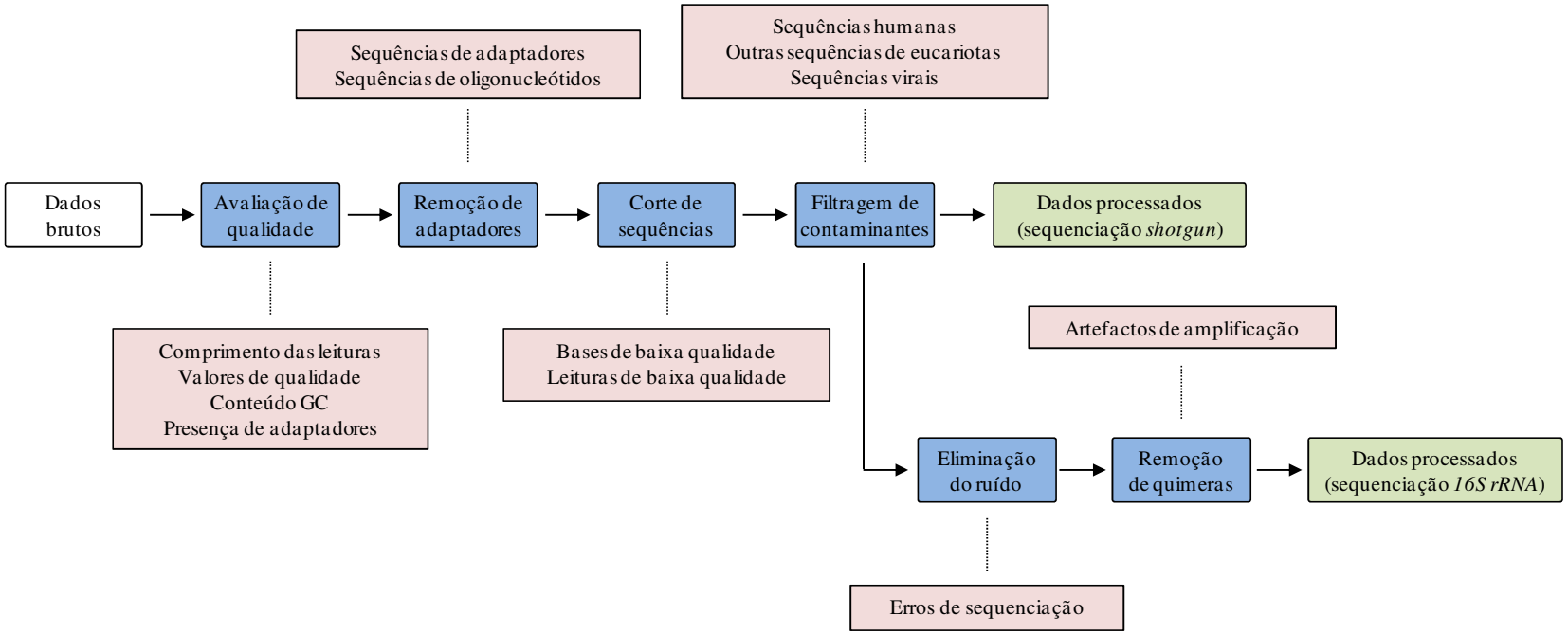

Figura 1.2. Diagrama representativo dos principais níveis de análise de qualidade e pré-processamento de leituras de sequenciação (análise primária) em metagenómica. O objectivo da análise primária é o de enriquecer os dados brutos da sequenciação em sequências com qualidade e representativas dos organismos biológicos de interesse.

Tabela 1.1. Programas para análise de qualidade e pré-processamento de leituras de sequenciação.

Programa	Ficheiros de entrada	Avaliação de qualidade	Remoção de adaptadores	Corte de qualidade	Eliminação de sequências contaminantes	Processamento de leituras emparelhadas	Relatório	Referência
AdapterRemoval v2	FASTQ	não	sim	sim	não	sim (a)	não	Schubert *et al.* (2016)
AlienTrimmer	FASTQ	não	sim	não	não	sim	não	Criscuolo and Brisse (2013)
Biostrings	FASTQ, FASTA	sim	sim	não	não	não	não	Pagès *et al.* (2017)
Btrim	FASTQ	não	sim	sim	não	não	não	Kong (2011)
Cutadapt	FASTQ, FASTA, .csfasta/.qual	não	sim	sim	não	não	não	Martin (2011)
DeconSeq	FASTQ, FASTA	não	não	não	sim	não	sim	Schmieder and Edwards (2011a)
FastQC	FASTQ, SAM, BAM	sim	não	não	não	não	sim	Andrews (2010)
FASTX-Toolkit	FASTQ, FASTA	sim	sim	sim	não	não	não	(b)
Flexbar	FASTQ, FASTA	não	sim	sim	não	sim	sim	Dodt *et al.* (2012)
htseq-qa **(c)**	FASTQ, SAM	sim	não	não	não	não	não	Anders *et al.* (2015)
Kraken Tools **(d)**	FASTQ	sim	sim	sim	não	sim	sim	Davis *et al.* (2013)
Meta-QC-Chain	FASTQ	sim	sim	sim	sim (e)	sim	sim	Zhou *et al.* (2014)
ngsShoRT	FASTQ, QSEQ	não	sim	sim	não	sim	sim	Chen *et al.* (2014)
PathoQC **(f)**	FASTQ, FASTA	sim	sim	sim	não	sim	sim	Hong *et al.* (2014a)
PEAT	FASTQ	não	sim	não	não	sim	não	Li *et al.* (2015)
PRINSEQ	FASTQ, FASTA+QUAL	sim	sim	sim	sim	não	sim	Schmieder and Edwards (2011b)
QC-Chain	FASTQ	sim	sim	sim	sim (g)	sim	sim	Zhou *et al.* (2013)
QTrim	FASTQ, FASTA+QUAL	sim	não	sim	não	não	não	Shrestha *et al.* (2014)
SAMstat	FASTQ, FASTA, SAM, BAM	sim	não	não	não	não	sim	Lassmann *et al.* (2011)
SeqPurge	FASTQ	sim	sim	sim	não	sim	não	Sturm *et al.* (2016)
Skewer	FASTQ	não	sim	sim	não	sim	não	Jiang *et al.* (2014)
Trim-Galore!	FASTQ	não	sim	sim	não	sim	não	Krueger (2012)
Trimmomatic	FASTQ	não	sim	sim	não	sim	não	Bolger *et al.* (2014)

(a) O programa *AdapterRemoval v2* permite fundir leituras emparelhadas de forma a criar uma leitura única; (b) FASTX-Toolkit: FASTQ/A short-reads pre-processing tools (2009); (c) O *htseq-qa* é um componente do pacote *HTSeq;* (d) Inclui os programas *Reaper* e *Minion*; (e) Apenas para sequências de eucariotas; (f) O *PathoQC* usa os programas *FastQC*, *Cutadapt* e *PRINSEQ* e é um módulo do programa *Pathoscope 2.0*; (g) Para sequências de procariotas e de eucariotas.

1.3.2. Corte de sequências adaptadoras e de baixa qualidade

Na sequenciação *shotgun*, as bibliotecas são compostas for fragmentos de tamanho variável, pelo que se o comprimento da leitura obtida no sequenciador for superior ao do respectivo inserto, a sequenciação progride para dentro da sequência adaptadora, ficando esta integrada na sequência da leitura. As sequências adaptadoras têm então de ser cortadas uma vez que podem interferir em passos subsequentes da análise, nomeadamente no alinhamento contra sequências de referência. A qualidade de uma base sequenciada é frequentemente avaliada através do *Phred score* (Ewing *et al.*, 1998; Ewing and Green, 1998), ou de um parâmetro equivalente. No caso do *Phred score*, o valor da qualidade está relacionado com a probabilidade de confiança da base atribuída ser a base real. Por exemplo, um valor de qualidade de 30 (Q30) corresponde a uma probabilidade de confiança de 99,9%. A remoção das bases de baixa qualidade é menos objectiva do que a remoção dos adaptadores, sendo comum que diferentes utilizadores sigam critérios distintos neste aspecto.

O *Cutadapt* é uma ferramenta que permite remover as sequências adaptadoras e efectuar cortes selectivos nas extremidades das leituras (Martin, 2011). O *FASTX-Toolkit* (FASTX-Toolkit: FASTQ/A short-reads pre-processing tools, 2009) é um outro conjunto de ferramentas que inclui, entre outros, a obtenção de estatísticas da qualidade e distribuição dos nucleótidos nas sequências (*FASTQ Information*), a remoção das sequências adaptadoras (*FASTQ/A Clipper*) e o corte de segmentos das sequências que apresentem menor qualidade (*FASTQ/A Quality Trimmer*). No caso de se trabalhar com dados obtidos em plataformas *Illumina*, particularmente no caso de leituras emparelhadas (que correspondem a 2 ficheiros *.fastq*), o *Trimmomatic* é uma alternativa a considerar (Bolger *et al.*, 2014). Outra possibilidade é usar o *package Biostrings* do *Bioconductor* (Pagès *et al.*, 2017) para utilizadores mais familiarizados com o *software R*. Este *package* contém várias ferramentas úteis para manipulação, contagem, transformação e "matching" de sequências, incluindo a possibilidade de efectuar o corte das extremidades das leituras. No entanto, se o utilizador requer uma análise mais adaptada ao seu projecto específico, poderá usar o *software HTSeq* (Anders *et al.*, 2015), que permite desenvolver facilmente *scripts* "personalizados" em *Python*, para as várias etapas do tratamento e análise de dados de sequenciação. Em certas situações, poderão não ser conhecidas as sequências completas dos adaptadores usados na preparação das bibliotecas, nomeadamente quando a sequenciação é encomendada a um prestador de serviços externo. Neste caso, os programas *Minion*, incluído no *software Kraken Tools* (Davis *et al.,* 2013), e *PEAT* (Li *et al.*, 2015), permitem detectar e remover sequências de adaptadores sem que estas necessitem de ser previamente introduzidas ou seleccionadas pelo utilizador.

Os passos de remoção de sequências adaptadoras e de baixa qualidade podem ser efectuados de forma sequencial e automatizada usando o *PathoQC* (Hong *et al.*, 2014a). Este *software* utiliza o *FastQC*, o *Cutadapt* e o *PRINSEQ* nesta ordem, por forma a gerar um conjunto de leituras de sequências pré-processadas de alta qualidade. Alguns programas permitem gerar um relatório com os resultados da análise de qualidade e/ou do processamento das leituras, o qual pode ser consultado mais tarde sem necessidade de se repetir todo o processo de análise e/ou tratamento dos dados.

1.3.3. Detecção e eliminação de sequências contaminantes

A etapa seguinte da análise primária corresponde à detecção e eliminação de sequências que não se pretende que estejam presentes no conjunto de dados. A título de exemplo, a colheita de uma amostra

do microbioma da cavidade bucal humana deverá conter, com elevada probabilidade, DNA de origem humana proveniente de células da mucosa bucal. Neste caso, as leituras que contêm sequências de fragmentos de DNA humano devem ser eliminadas antes da análise secundária. Para este fim, existem pelo menos 4 ferramentas disponíveis, que incluem o *PRINSEQ*, o *DeconSeq*, o *Meta-QC-Chain* (Zhou *et al.*, 2014) e o *QC-Chain* (Zhou *et al.*, 2013), as quais usam diferentes abordagens para detectar a presença de sequências contaminantes. O *PRINSEQ* faz o cálculo do *dinucleotide odds ratio* como forma de detectar dinucleótidos sugestivos da presença de sequências de organismos contaminantes (Schmieder and Edwards, 2011b), enquanto o *DeconSeq* efectua uma pesquisa de sequências contaminantes usando bases de dados específicas (Schmieder and Edwards, 2011a). Esta pesquisa permite, por exemplo, reter as bases de dados que possam revelar sequências semelhantes entre vírus e humanos, e remover as que têm apenas sequências de origem humana. O *QC-Chain* utiliza uma dupla abordagem de pesquisa em bases de dados para detectar e quantificar as sequências contaminantes. Numa primeira fase, o ficheiro de leituras é analisado no programa *Paralell-META* (Su *et al.*, 2012a; Su *et al.*, 2014a), o qual extrai as sequências do gene *16S rRNA* dos procariotas e do gene *18S rRNA* dos eucariotas, se existentes, e alinha-as contra várias bases de dados, permitindo identificar leituras que não se esperam encontrar na amostra sequenciada. Numa segunda fase, é seleccionada uma proporção de leituras aleatórias, a partir do conjunto de dados original, as quais são alinhadas contra a base de dados de sequências NCBI-*nucleotide* (*nt*) (National Center for Biotechnology Information - Nucleotide, 1988) usando o *BLASTn* (Altschul *et al.*, 1990). A informação assim obtida pode servir de confirmação dos resultados obtidos na primeira abordagem e também fornecer uma estimativa da proporção de contaminação nos dados originais. As leituras contaminantes podem ser filtradas através do alinhamento contras as sequências dos genomas das espécies contaminantes usando o *software* de alinhamento *Bowtie* (Langmead *et al.*, 2009). Finalmente, no caso do *Meta-QC-Chain*, que é o único programa criado especificamente para avaliação de qualidade em metagenómica (Zhou *et al.*, 2014), as leituras do gene *18S rRNA* são pesquisadas e extraídas dos dados originais, e alinhadas contra a base de dados *SILVA* (Quast *et al.*, 2013, Yilmaz *et al.*, 2014). As leituras podem depois ser filtradas com um programa de alinhamento escolhido pelo utilizador.

1.3.4. Detecção de erros de sequenciação

Os erros de sequenciação ocorrem em todas as plataformas tecnológicas e poderão resultar na classificação de leituras originalmente idênticas em grupos taxonómicos distintos, aumentando de forma artificial a diversidade microbiológica de uma dada amostra, ou em novos grupos taxonómicos que não estão presentes na amostra original. No caso da metagenómica, há que ter em conta que os dados de sequenciação *shotgun* podem apresentar um perfil de erros distinto do obtido com a sequenciação de amplicões do gene *16S rRNA* (Bokulich *et al.*, 2013). Consequentemente, as tarefas de pré-processamento de leituras devem ser adaptadas ao método em causa a fim de se evitar a sobre-estimação da diversidade microbiana. A análise e detecção de erros de sequenciação podem ser efectuadas com diferentes programas, dos quais alguns exemplos estão referidos na **tabela 1.2**. No caso da sequenciação de amplicões, Quince *et al.* (2009) desenvolveram um algoritmo designado *PyroNoise*, cuja função é a de remover o "ruído" (*denoising*) das leituras do equipamento 454/*Roche*, permitindo obter uma estimativa mais precisa do número de unidades taxonómicas operacionais (UTO(s)) existentes na amostra. No caso do equipamento *MiSeq* da *Illumina*, Mysara *et al.* (2016) propuseram um algoritmo baseado em *machine-learning*, que detecta potenciais posições erróneas em leituras emparelhadas obtidas a partir de amplicões do gene *16S rRNA*. No mesmo âmbito, foram propostas linhas de orientação para exclusão de leituras de baixa qualidade destes amplicões e de

limiares de exclusão/aceitação de UTO(s) para plataformas *Illumina* (Bokulich *et al.*, 2013). Puente-Sánchez *et al.* (2016) desenvolveram um outro algoritmo, denominado *Poisson binomial filtering* (PBF), que calcula uma distribuição da probabilidade de erro de uma sequência baseada nos valores de qualidade das respectivas bases individuais, por forma a filtrar leituras contendo potenciais erros ocorridos no processo de sequenciação de amplicões. Uma mais-valia deste método é o de se basear apenas nos valores de qualidade das bases individuais obtidos durante o *base-calling*, o que permite a sua aplicação independentemente da plataforma tecnológica de sequenciação utilizada (Puente-Sánchez *et al.*, 2016).

Tabela 1.2. Programas de remoção de erros de sequenciação.

Programa	Referência	Método	URL
IPED	Mysara *et al.* (2016)	Algoritmo de *machine-learning*	http://science.sckcen.be/en/Institutes/EHS/MCB/MIC/Bioinformatics/IPED
PBF	Puente-Sánchez *et al.* (2016)	Distribuição de Poisson com base em valores de qualidade	https://github.com/fpusan/moira
PyroNoise	Quince *et al.* (2009)	Modelo de mistura	https://code.google.com/archive/p/ampliconnoise/downloads

1.3.5. Remoção de sequências quiméricas

A análise primária deve ainda incluir, na sua etapa final, a detecção e a remoção de sequências quiméricas. Estas sequências artificiais, que resultam de recombinação de fragmentos de DNA *in vitro*, podem formar-se durante a reacção de amplificação do gene *16S rRNA* ou de outros fragmentos. Este fenómeno foi bem ilustrado por Bradley and Hillis (1997) ao amplificarem alelos de genes de cópia única em indivíduos heterozigóticos. As sequências quiméricas resultam da junção de 2 ou mais produtos da PCR cuja extensão termina precocemente, e que podem actuar como oligonucleótidos iniciadores nos passos subsequentes da reacção. Se os produtos truncados contiverem na sua sequência uma base variante relativamente às sequências com as quais emparelham, irão introduzir esta nova base no fragmento amplificado. De acordo com Bradley and Hillis (1997), se a restante porção da sequência amplificada contiver uma outra base variante, então o produto quimérico irá conter 2 bases variantes que não existiam em qualquer das sequências genómicas na amostra original. As quimeras não detectadas podem ser interpretadas como novas espécies, o que pode afectar a estimativa da diversidade de uma comunidade de microorganismos e a comparação entre populações (Edgar *et al.*, 2011).

Tabela 1.3. Programas de remoção de sequências quiméricas.

Programa	Referência	Método	URL
CATCh	Mysara *et al.* (2015)	Algoritmo de *machine-learning* que combina os resultados de diferentes programas de análise de sequências quiméricas	http://science.sckcen.be/en/Institutes/EHS/MCB/MIC/Bioinformatics/CATCh
DECIPHER	Wright *et al.* (2012)	Detecção de pequenos fragmentos (30 nt) em diferentes grupos filogenéticos	http://decipher.cee.wisc.edu/index.html
Perseus	Quince *et al.* (2011)	Modelo de mistura baseado no agrupamento de *flowgrams* do 454 (Roche)	https://code.google.com/archive/p/ampliconnoise/
UCHIME	Edgar *et al.* (2011)	Comparação da sequência das leituras com sequências parentais em bases de dados de referência ou comparação das abundâncias das respetivas sequências	http://drive5.com/usearch/manual/uchime_algo.html

A detecção e eliminação das sequências quiméricas podem ser efectuadas usando os programas descritos na **tabela 1.3**. Os programas *UCHIME* (Edgar *et al.*, 2011) e *Perseus* (Quince *et al.*, 2011) são de longe os mais referenciados na literatura, embora no caso do *Perseus* este seja apenas aplicável

a amplicões gerados pela plataforma de sequenciação 454/*Roche*. O programa *UCHIME* efectua a deteção de quimeras usando bases de dados de sequências "livres" de quimeras, ou através de dados de abundância (Edgar *et al.*, 2011), enquanto o *Perseus* utiliza somente dados de abundância de sequências (Quince *et al.*, 2011). Uma das formas de estimar o erro associado à formação de fragmentos quiméricos, consiste em analisar os fragmentos amplificados e sequenciados a partir de amostras com diversidade previamente conhecida (Quince *et al.*, 2011).

1.4. Classificação taxonómica de leituras de sequenciação *shotgun*

O objectivo essencial de qualquer estudo de metagenómica é o de conhecer a origem taxonómica de cada sequência de DNA presente numa dada amostra. Este objectivo pode ser concretizado através da **classificação taxonómica** (ou *binning*), que pode ser descrito como o processo de atribuir uma dada leitura de sequenciação a uma unidade taxonómica definida (*bin*). Se um *bin* representar um táxon de baixo nível como uma espécie, o conjunto de leituras atribuídas a cada *bin* individual poderá ser usado para reconstruir o genoma de cada espécie presente na comunidade de microorganismos. A selecção do método utilizado para efectuar a classificação taxonómica é um aspecto muito importante, a que não é alheio o facto de existir actualmente um número muito elevado de métodos e programas disponíveis para este fim. No entanto, no meio da aparente diversidade, é possível categorizar os métodos de classificação taxonómica em 4 tipos principais, que incluem os métodos de semelhança, composição, híbridos e de genes marcadores. Todos estes métodos, assim como os métodos de montagem ou de abundância, descritos mais à frente, incluem-se na chamada "análise secundária", e são normalmente implementados nas linguagens C/C++, *Perl*, *Python* ou *Matlab*. A **figura 1.3** apresenta um diagrama dos principais métodos de análise secundária para dados de sequenciação *shotgun*, onde se evidenciam alguns dos programas disponíveis.

1.4.1. Métodos de semelhança

Tal como o nome sugere, estes métodos baseiam-se na pesquisa de semelhança e no alinhamento entre a sequência das leituras e as sequências disponíveis nas bases de dados de genomas de referência, um processo que é designado por *fragment recruitment* (Rusch *et al.*, 2007). Uma vez que o alinhamento das leituras está dependente do conhecimento prévio das respectivas sequências de referência, estes métodos designam-se por **métodos supervisionados** (Rosen *et al.*, 2008). Existem vários programas de alinhamento de sequências, entre os quais o *Bowtie* (Langmead *et al.*, 2009), *Burrows-Wheeler Transform/Burrows-Wheeler Alignment* (*BWT/BWA*, Li and Durbin, 2009), *FR-HIT* (Niu *et al.*, 2011), *BLAT* (Kent, 2002), *MegaBLAST* (Zhang *et al.*, 2000), *BLAST* (Altschul *et al.*, 1990), *LAST* (Kielbasa *et al.*, 2011) e *SOAP2* (Li *et al.*, 2009). No caso do *BLAST*, um dos programas de alinhamento mais populares, as leituras de sequenciação são alinhadas directamente com sequências de genomas de referência, e as leituras alinhadas são atribuídas à árvore taxonómica usando o melhor resultado (*best hit*) ou o conjunto de melhores resultados, de acordo com as estatísticas *bit-score* ou *expect-value*. No entanto, este alinhamento só produz bons resultado se na(s) base(s) de dados pesquisada(s) existirem sequências com homologia muito próxima para comparação. Para evitar uma atribuição incorrecta das leituras, poderá aumentar-se a estringência de diferentes parâmetros do alinhamento, mas neste caso o mais provável é que se perca homologia com qualquer das sequências na base de dados (Ghosh *et al.*, 2012).

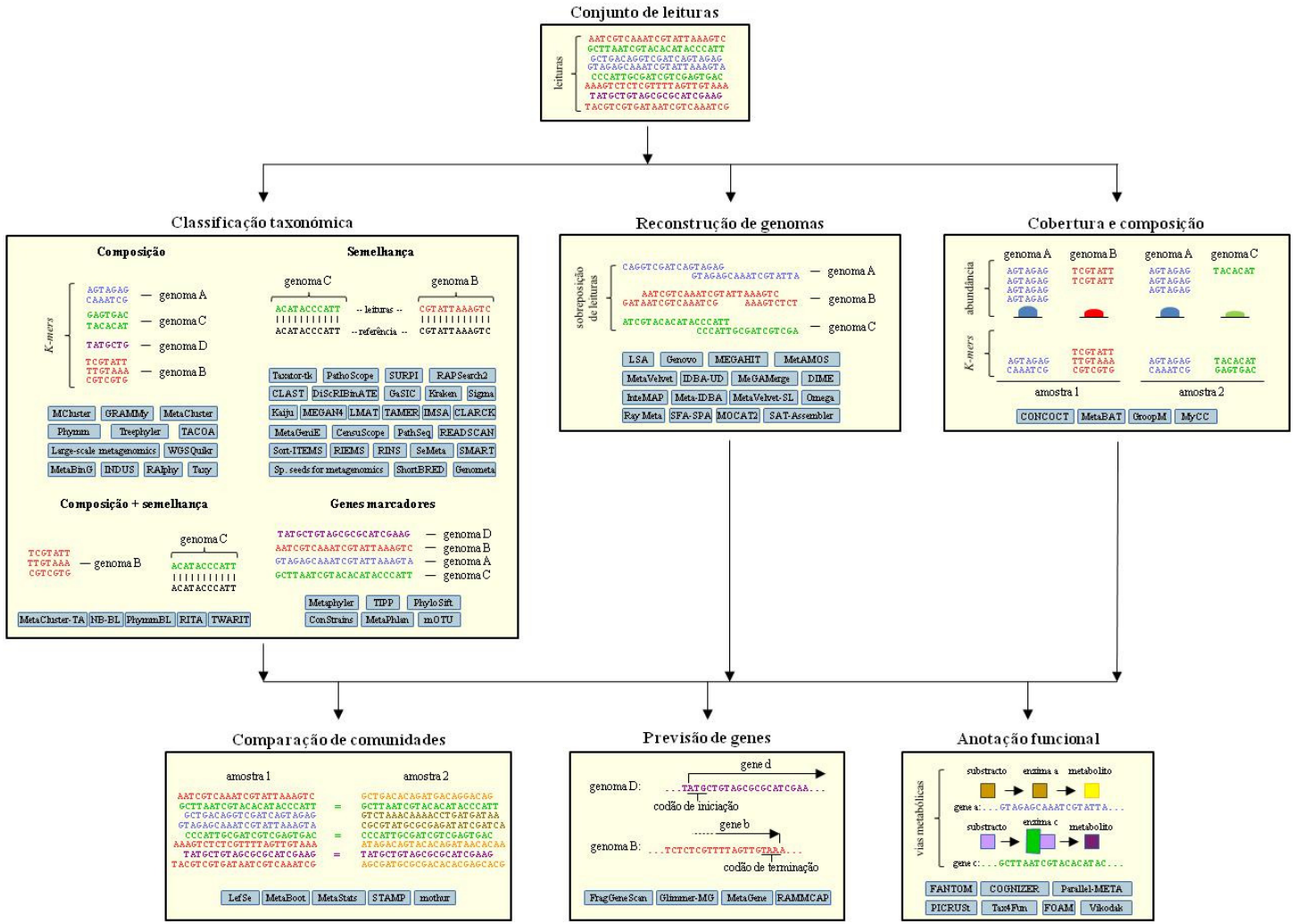

Figura 1.3. Diagrama representativo dos vários níveis da análise secundária de dados de sequenciação metagenómica *shotgun*. Os objectivos desta análise são o de determinar a composição da comunidade de microorganismos, conhecer o respectivo perfil funcional, identificar novos genes e comparar as características da comunidade com outras comunidades amostradas em outros locais ou em alturas difererentes. Alguns dos programas que podem ser usados em cada nível de análise estão indicados nas caixas azuis.

Na **tabela 1.4** estão listados alguns programas para classificação taxonómica de leituras de sequenciação metagenómica *shotgun* por métodos de semelhança. Alguns destes programas utilizam os algoritmos do *BLAST* para fazer o alinhamento de leituras contra as bases de dados. O *software Metagenome Analyser 4* (*MEGAN4*) é um programa muito citado e que compara conjuntos de sequências com as bases de dados de sequências proteícas *non-redundant protein sequences* (*nr*) e *nt* do *NCBI*, entre outras, ou com sequências de genomas completos, usando por exemplo o *BLASTX* (Altschul *et al.*, 1990). De seguida, é feita uma classificação de cada uma das sequências num *táxon* do *NCBI taxonomy* ou da base de dados *SILVA* com base no menor ancestral comum (Huson *et al.*, 2007). Ao contrário de outros métodos baseados na procura de homologias, o *software Taxonomic Assignment of Metagenomic Sequence Reads* (*TAMER*) é um programa que faz a classificação de leituras sem as atribuir uma a uma à árvore taxonómica (Jiang *et al.*, 2012). O programa usa o *MegaBLAST* para pesquisar sequências homólogas na base de dados *nt* e modela estatisticamente a probabilidade de encontrar as diferentes leituras no conjunto de dados, usando como parâmetros conhecidos o comprimento máximo do alinhamento e o número de bases emparelhadas de cada leitura, para cada genoma de referência em que as leituras foram alinhadas. As leituras são atribuídas à árvore taxonómica de acordo com o resultado de uma função que calcula a probabilidade de uma determinada leitura ser gerada por um dado genoma (Jiang *et al.*, 2012).

Os métodos de alinhamento apresentam limitações importantes quando trabalham com grandes volumes de dados, uma vez que o número de leituras e o tempo de computação estão linearmente correlacionados (Scheuch *et al.*, 2015). A pesquisa de homologia de milhões de leituras individuais, num espaço de procura muito elevado, composto por milhares de sequências genómicas, tem como consequência um tempo de processamento muito longo. Para contornar este obstáculo têm sido implementadas soluções que envolvem a divisão dos dados por múltiplos servidores em paralelo (Rawat *et al.*, 2014). No entanto, uma forma bastante simples de ultrapassar este problema e que não requer a existência de grandes recursos computacionais, é a utilizada pelo programa *CensuScope* (Shamsaddini *et al.*, 2014). Este programa amostra de forma aleatória um reduzido número de leituras do conjunto de dados original, que pode ser definido pelo utilizador, e alinha estas leituras usando por exemplo o *BLAST* contra as bases de dados do *nt* ou *MetaPhLan* (Segata *et al.*, 2012). Os *best hits* são utilizados para obter a informação taxonómica do *NCBI Taxonomy*. Este processo é repetido de acordo com um determinado número de iterações com substituição, obtendo-se no final uma amostragem representativa dos grupos taxonómicos presentes na amostra e uma estimativa da abundância de cada um destes. O utilizador pode definir um limite de exclusão para evitar que determinados grupos taxonómicos, presentes a níveis diminutos nos dados originais, possam ser seleccionados no processo de amostragem e, como tal, contribuam para uma sobre-representação desses grupos no conjunto de dados (Shamsaddini *et al.*, 2014).

1.4.2. Métodos de composição

Os métodos de composição (ou composicionais) são um outro tipo de métodos de "binning", mas que não se baseiam na pesquisa de homologia de sequências e alinhamentos. Estes métodos usam características intrínsecas da sequência de DNA de cada leitura, ou de cada *contig*, para comparar com as mesmas características obtidas a partir das sequências genómicas de referência (métodos supervisionados), ou para agrupar as leituras em *bins* de acordo com características semelhantes entre si, sendo então designados por **métodos não supervisionados**. As características que permitem diferenciar as leituras de sequenciação de uma amostra, constituída por vários genomas, incluem entre outras a frequência de oligonucleótidos de 2-15 pb, designados como *k-mers* ou *n-mers*, contíguos ou

Tabela 1.4. Programas de análise de leituras de sequenciação metagenómica *shotgun* por métodos de semelhança.

Programa	Referência	Algoritmo/programa de alinhamento	Bases de dados de sequências de referência/taxonomia	URL
CensuScope	Shamsaddini *et al.* (2014)	*BLAST*, *bowtie2* (a), *BWA*	*NCBI-nt, MetaPhLan/NCBI Taxonomy*	https://hive.biochemistry.gwu.edu/dna.cgi?cmd=censuscope
CLARK	Ounit *et al.* (2015)	Frequência de *k-mers*	*NCBI-RefSeq* (c)/na	http://clark.cs.ucr.edu/
CLAST	Yano *et al.* (2014)	Alinhamento exacto de *k-mers* e alinhamento em banda	*NCBI RefSeq/NCBI Taxonomy*	https://github.com/masayano/CLAST
DiScRIBinATE	Ghosh *et al.* (2010)	*BLASTX*	*NCBI-nr/NCBI Taxonomy*	http://metagenomics.atc.tcs.com/binning/DiScRIBinATE/
GaSIC	Lindner and Renard (2013)	(b)	*NCBI-RefSeq*/ni	https://sourceforge.net/projects/gasic/
Genometa	Davenport *et al.* (2012)	*bowtie*	(b)/na	http://genomics1.mh-hannover.de/genometa/index.php?Site=Home
IMSA	Dimon *et al.* (2013)	*BLASTn*, *bowtie*, *BLAT*	*NCBI-nt/NCBI-nt*	https://sourceforge.net/projects/arron-imsa/?source=directory
Kaiju	Menzel *et al.* (2016)	*BWT*	*NCBI-RefSeq, NCBI-nr/NCBI Taxonomy*	http://kaiju.binf.ku.dk/
Kraken	Wood and Salzberg (2014)	Alinhamento exacto de *k-mers*	*NCBI-RefSeq* ou (b)/ni	http://ccb.jhu.edu/*software*/kraken/
LMAT	Ames *et al.* (2013)	Alinhamento exacto de *k-mers*	*NCBI genome database of microbial genomes/NCBI Taxonomy*	http://computation.llnl.gov/projects/livermore-metagenomics-analysis-toolkit
MEGAN4	Huson *et al.* (2011)	*BLASTX*	*NCBI-nt, NCBI-nr, NCBI-RefSeq/NCBI Taxonomy, SILVA database*	http://ab.inf.uni-tuebingen.de/software/megan4/
MetaGeniE	Rawat *et al.* (2014)	*BWT*, *BLAT*	*NCBI complete bacterial genomes, NCBI-RefSeq*/ni	https://github.com/ngsclinical/metagenie
PathoScope	Hong *et al.* (2014b)	*bowtie2*	(b)/*NCBI Taxonomy, PathoDB*	https://sourceforge.net/projects/pathoscope/
PathSeq (d)	Kostic *et al.* (2011)	*BLASTN*, *BLASTX*	*NCBI complete bacterial genomes, NCBI-nt* (vírus e fungos) e *NCBI-nr*/ni	http://www.broadinstitute.org/*software*/pathseq/
RAPSearch2	Zhao *et al.* (2012)	Alinhamento exacto de aa	*NCBI-nr, eggNOG* (e), *IMG 3.0* (f)/na	http://omics.informatics.indiana.edu/mg/RAPSearch2/
READSCAN	Naeem *et al.* (2013)	*SMALT* (g)	*NCBI RefSeq/NCBI Taxonomy*	http://cbrc.kaust.edu.sa/readscan/
RIEMS	Scheuch *et al.* (2015)	*MegaBLAST*, *BLASTn*	*NCBI-nt/NCBI Taxonomy*	https://www.fli.de/de/404/
RINS	Bhaduri *et al.* (2012)	*BLAT*	*GenBank* /*ICTV* (h)	http://khavarilab.stanford.edu/tools-1/#tools
SeMeta	Le *et al.* (2016)	*BLASTX*	*Protein RefSeq/NCBI Taxonomy*	http://it.hcmute.edu.vn/bioinfo/metapro/SeMeta.html
ShortBRED	Kaminski *et al.* (2015)	*USEARCH*	*Protein marker database*/na	http://huttenhower.sph.harvard.edu/shortbred
Sigma	Ahn *et al.* (2015)	*bowtie2*	*RefSeq genomes*/ni	http://sigma.omicsbio.org/
SMART	Lee *et al.* (2016)	Alinhamento exacto de *k-mers*	*NCBI GenBank/NCBI Taxonomy*	https://bitbucket.org/ayl/smart
Sort-ITEMS	Monzoorul *et al.* (2009)	*BLASTX*	*NCBI-nr/NCBI Taxonomy*	http://metagenomics.atc.tcs.com/binning/SOrt-ITEMS/
Spaced seeds for metagenomics (i)	Břinda *et al.* (2015)	Alinhamento espaçado de *k-mers*	*NCBI-RefSeq*/ni	http://github.com/gregorykucherov/spaced-seeds-for-metagenomics
SURPI	Naccache *et al.* (2014)	*SNAP* (j), *RAPSearch2*	*NCBI-nt, NCBI-nr/NCBI Taxonomy*	https://github.com/chiulab/surpi
TAMER	Jiang *et al.* (2012)	*MegaBLAST*	*NCBI-nt/NCBI Taxonomy*	http://faculty.wcas.northwestern.edu/~hji403/MetaR.htm
Taxator-tk	Dröge *et al.* (2015)	*BLASTn*, *MegaBLAST*, *tBLASTX*, *LAST*	*Microbial RefSeq/NCBI Taxonomy*	https://github.com/fungs/taxator-tk/

Abreviaturas: na, não aplicável; ni, não indicado. (a) Langmead and Salzberg (2012); (b) escolhido pelo utilizador; (c) Pruitt *et al.* (2012); (d) permite também fazer a montagem das leituras usando o programa *Velvet* (Zerbino and Burney, 2008); (e) http://eggnog.embl.de/; (f) conjunto de sequências não redundantes (98%) recolhidas a partir de http://img.jgi.doe.gov/; (g) www.sanger.ac.uk/science/tools/smalt-0; (h) *International Committe on Taxonomy of Viruses*; (i) representa uma modificação do *software Kraken* para incluir alinhamento espaçado de *k-mers*; (j) Zaharia *et al.* (2011).

descontínuos (Ma *et al.*, 2002), a presença de *strings* longas (*w-mers*) (Wang *et al.*, 2012), e os codões sinónimos (Rho *et al.*, 2010). Os trabalhos de Karlin *et al.* (1997) demonstraram que as frequências de dinucleótidos ou de tetranucleótidos diferentes tendem a variar menos nos genomas de uma mesma espécie do que entre genomas de espécies diferentes, o que serve de base à aplicação dos métodos composicionais nos estudos de metagenómica. A composição dos *k-mers* assume particular relevo na classificação taxonómica uma vez que é assumido que estas sequências "transportam" um sinal filogenético (Campbell *et al.*, 1999). No entanto, em consequência de variações da sequência genómica a nível local, os métodos de composição baseados na frequência de *k-mers* não funcionam tão bem para sequências inferiores a 500 pb (Bentley and Parkhill, 2004; Prabhakara and Acharya, 2012). Adicionalmente, também não é expectável que produzam bons resultados na diferenciação de genomas muito "próximos", como sejam os genomas das diferentes estirpes de uma mesma espécie. Os *w-mers* (w>=36) podem servir para identificar uma espécie única (Fofanov *et al.*, 2004) mas, devido ao seu comprimento, têm pouca aplicação na diferenciação de sequências curtas, que correspondem à maioria das leituras obtidas nas plataformas de sequenciação actuais. Na **tabela 1.5** estão descritos alguns *softwares* disponíveis para análise de dados por métodos de composição.

Tabela 1.5. Programas de análise de leituras de sequenciação metagenómica *shotgun* por métodos de composição.

Programa	Referência	Tipo	Método	URL
GRAMMy (a)	Xia *et al.* (2011)	S	Modelo de mistura	http://meta.usc.edu/softs/grammy/
INDUS	Mohammed *et al.* (2011a)	S	Frequência *k-mers* (fragmentos genómicos)	http://metagenomics.atc.tcs.com/INDUS/
Large-scale metagenomics	Vervier *et al.* (2016)	S	frequência k-mers (k={4,6,8,10,12}), machine learning	http://cbio.mines-paristech.fr/largescalemetagenomics/
MCluster	Liao *et al.* (2014)	NS	*N-grams*, agrupamento *K-means*	https://github.com/zhangruichang/MCluster_VS2010
MetaBinG	Jia *et al.* (2011)	S	Modelos de Markov de 5ª ordem	http://cbb.sjtu.edu.cn/~ccwei/pub/*software*/MetaBinG/MetaBinG.php
MetaCluster	Wang *et al.* (2012)	NS	Agrupamento com base em *w-mers* comuns; cálculo da distribuição de *q-mers* nos agrupamentos; construção de *clusters* com o algoritmo *k-means*	http://i.cs.hku.hk/~alse/MetaCluster/
Phymm	Brady and Salzberg (2009)	S	Frequência *k-mers* (k=1-12 bases), modelos de Markov de ordem variável	http://www.cbcb.umd.edu/software/phymm/
RAIphy	Nalbantoglou *et al.* (2011)	S	Frequência *k-mers*, índice de abundância relativa por cada *k-mer*	http://bioinfo.unl.edu/raiphy.php
TACOA	Diaz *et al.* (2009)	S	*K-nearest neighbour*	http://www.cebitec.uni-bielefeld.de/index.php/2-uncategorised/99-tacoa?highlight=WyJ0YWNvYSJd
Taxy	Meinicke *et al.* (2011)	S	Frequência *k-mers* (k=6-8) na amostra completa; perfil taxonómico é derivado de um modelo de mistura	http://gobics.de/TaxyPro/
Treephyler	Schreiber *et al.* (2010)	S	HMM, domínios famílias PFAM, classifica em árvore	http://www.gobics.de/fabian/treephyler.php
WGSQuikr	Koslicki *et al.* (2014)	S	Frequência *k-mers* (k~7)	https://sourceforge.net/projects/wgsquikr/

Abreviaturas: S, método supervisionado; NS, método não supervisionado. (a) O *GRAMMy* também permite classificar as leituras de sequenciação por métodos de semelhança usando o *BLAT* ou o *BLAST*, por exemplo.

O *Phymm* é um exemplo de *software* muito utilizado que utiliza sequências de oligonucleótidos de tamanho variável para caracterizar o genoma de cada espécie (Brady and Salzberg, 2009). O algoritmo é baseado em *interpolated Markov models* (modelos de Markov de ordem variável, MMOV), que são uma forma de cadeia de Markov em que a probabilidade do próximo estado é computada usando a informação de um número variável de estados anteriores. Os cromossomas e plasmídeos da base de dados de sequências de referência *RefSeq* (O´Leary *et al.*, 2016; Tatusova *et al.*, 2016) do NCBI foram usados para treinar os MMOV do *Phymm*. O treino dos modelos consistiu na construção de distribuições de probabilidade que representam os padrões de oligonucleótidos com 1-12 bases de comprimento que melhor caracterizam o genoma de cada microorganismo. Para classificar os fragmentos dos conjuntos de dados de metagenómica, cada MMOV analisa uma sequência de DNA da amostra, devolvendo um valor que corresponde à probabilidade do MMOV ter gerado aquela sequência. Por sua vez, este valor serve como estimativa da probabilidade de que essa sequência pertence ao conjunto das sequências de referência em que o MMOV foi treinado (Brady and Salzberg, 2009). Por comparação com o *BLAST*, que se baseia em alinhamento, os autores verificaram que o *Phymm* só era superior na classificação nos níveis de classe e filo para as leituras de maior comprimento (Brady and Salzberg, 2009), sugerindo que o sucesso da classificação taxonómica poderia aumentar combinando o método de composição *Phymm* com um método de alinhamento. No programa *Grammy* (Xia *et al.*, 2011), é possível optar por métodos composicionais, que estimam a abundância relativa de cada genoma através de um algoritmo de *expectation-maximization* (EM) a partir da frequência de *k-mers*, ou de semelhança, usando o *BLAST* ou o *BLAT*, para analisar um mesmo conjunto de dados.

1.4.3. Métodos híbridos

Os métodos híbridos combinam algoritmos dos tipos composicional e de semelhança para a classificação de leituras, numa tentativa de se conseguir obter "o melhor de dois mundos" (**tabela 1.6**). Brady and Salzberg (2009) desenvolveram também uma abordagem mista para análise de dados de metagenómica que combina o *Phymm* com o *BLAST*. O algoritmo híbrido *PhymmBL* produziu melhores resultados para todos os comprimentos de leituras e níveis taxonómicos considerados, quando comparado com o *BLAST* ou o *Phymm* individualmente. O *PhymmBL* utiliza uma função que combina o valor obtido para o melhor MMOV com o valor mais baixo do *expect-value* devolvido pelo *BLAST*, ou seja, maximiza a classificação taxonómica ao ter em consideração os valores mais significativos de ambos os algoritmos. Em 2011, foi publicada uma nova versão do algoritmo *PhymmBL* que permite obter um valor de confiança para prever a taxonomia atribuída a cada leitura (Brady and Salzberg, 2011). Uma abordagem similar à do *PhymmBL* foi seguida por Parks *et al.* (2011), que desenvolveram o método híbrido *Naïve Bayes-BLAST* (*NB-BL*). Este método consiste numa extensão do algoritmo de classificação composicional *Naïve Bayes* (*NB*), desenvolvido por Sandberg *et al.* (2001), em conjunto com o *BLAST*, para efectuar uma classificação específica de nível taxonómico.

Uma forma distinta de aplicar métodos composicionais e de semelhança em paralelo foi usada no *software MetaCluster-TA* (Wang *et al.*, 2014). Neste caso, as leituras são primeiramente colocadas em agrupamentos de acordo com *w-mers* (com *w*>35). As leituras de cada agrupamento são sujeitas a montagem (*assembly*) usando o *software IDBA-UD* (Peng *et al.*, 2012), para formar *contigs* de tamanho >500 pb em que pelo menos 95% das bases são iguais entre leituras. A ideia deste método híbrido é a de que as leituras alinhadas com o mesmo *contig* após montagem têm uma maior probabilidade de terem sido amostradas da mesma espécie original (Wang *et al.*, 2014). Os *contigs* são

depois novamente agrupados com base em *q-mers* (com *q* variável em função do comprimento do *contig*), usando o algoritmo de agrupamento *K-means*, uma vez que a existência de distribuições semelhantes de *q-mers* é uma indicação de que as leituras provêm do mesmo genoma. Por fim, os *contigs* de cada agrupamento são alinhados contra os genomas de referência usando o *BLASTn* e anotados usando o menor ancestral comum (MAC) de todos os genomas alinhados.

Tabela 1.6. Programas de análise de leituras de sequenciação metagenómica *shotgun* por métodos híbridos.

Programa	Referência	Método de composição/semelhança	URL
MetaCluster-TA	Wang *et al.* (2014)	*w-mers* (*w*>35)/*BLASTn*	http://i.cs.hku.hk/~alse/MetaCluster/
NB-BL	Parks *et al.* (2011)	Classificador NB com base em *n-mers* (*n*=10)/*BLAST*	http://kiwi.cs.dal.ca/*Software*/FCP
PhymmBL	Brady and Salzberg (2009), Brady and Salzberg (2011)	Modelos de Markov de ordem variável/*BLAST*	http://www.cbcb.umd.edu/*software*/phymm/
RITA	MacDonald *et al.* (2012)	Classificador NB com base em *n-mers* (*n*=10)/ *discontiguous MegaBLAST*, *BLASTN*, *BLASTX*	http://kiwi.cs.dal.ca/*Software*/RITA
TWARIT	Reddy *et al.* (2012)	*n-mers* (n=4)/*BWA*	http://metagenomics.atc.tcs.com/Twarit/

1.4.4. Métodos de genes marcadores

Os métodos de genes marcadores usam apenas uma colecção parcial de genes dos genomas de referência para realizarem a classificação taxonómica, sendo por isso computacionalmente mais rápidos do que os métodos de semelhança ou composição (**tabela 1.7**). A classificação das leituras é efectuada através da pesquisa de homologia na colecção de sequências, usando por exemplo o *BLASTn*. O *software Metagenomic Phylogenetic Analysis* (*MetaPhlAn*) é o método de genes marcadores mais citado, que utiliza um conjunto pré-seleccionado de sequências marcadoras dos domínios *Bacteria* e *Archaea* (média de 231 sequências marcadoras por espécie), contra as quais são mapeadas as leituras da sequenciação *shotgun* (Segata *et al.*, 2012). As sequências marcadoras são obtidas a partir de sequências codificantes extraídas dos genomas existentes na base de dados *Integrated Microbial Genomes* (IMG), as quais foram seleccionadas de acordo com critérios biológicos específicos e de qualidade, por forma a serem representativas do nível taxonómico de espécie ou superior (Markowitz *et al.*, 2010; Markowitz *et al.*, 2012). Por exemplo, o gene *16S rRNA* ou genes com mais do que 2 cópias por genoma foram excluídos da lista final de sequências por não poderem ser univocamente atribuíveis a um único *clade* (i.e., um grupo de organismos que se julga englobar todos os descendentes de um ancestral comum). Uma vez que a especificidade das sequências marcadoras é muito elevada, não há obrigatoriedade de se efectuar um pré-processamento de leituras pois a probabilidade de existirem leituras a alinharem por acaso, numa dada sequência da colecção, é muito baixa.

Dado ter por base um alinhamento de sequências, o *MetaPhlAn* não consegue classificar em *clades* conhecidos as leituras que pertencem a genomas que ainda não foram sequenciados, pelo que aquelas são colocadas num *clade* "não classificado" do ancestral comum, para o qual existe informação de sequência disponível (Segata *et al.*, 2012). Uma outra desvantagem do programa é que um determinado gene pode existir em vários genomas que compõem um dado *clade*, mas pode estar ausente em outros microorganismos pertencentes a esse *clade* que ainda estão por sequenciar, fazendo com que a sequência marcadora não seja específica daquele *clade* (Segata *et al.*, 2012). Inversamente, um gene de um microorganismo ainda não sequenciado, pertencente a um novo *clade*, pode ter a mesma sequência de um outro *clade* conhecido que esteja incluído na colecção utilizada pelo

MetaPhlAn. Neste caso, as leituras originárias do genoma desconhecido serão colocadas no *clade* conhecido, perdendo-se assim informação relevante sobre a diversidade da comunidade de microorganismos. No entanto, é expectável que a colecção de sequências do *MetaPhlAn* vá sendo actualizada de forma periódica, em consonância com as actualizações da bases de dados IMG, o que tenderá a diminuir gradualmente o problema da identificação de novas espécies ou estirpes. A versão mais recente deste *software*, designada *MetaPhlAn2*, permite já a identificação de estirpes e a quantificação de sequências de origem eucariota e viral (Truong *et al.*, 2015).

Tabela 1.7. Programas de análise de leituras de sequenciação metagenómica *shotgun* por métodos de genes marcadores.

Programa	Referência	Genes/sequências marcadoras	URL
ConStrains	Luo *et al.* (2015)	231 sequências marcadoras/espécie (1221 espécies)	https://bitbucket.org/luo-chengwei/constrains
MetaPhlAn	Segata *et al.* (2012)	231 sequências marcadoras/espécie (1221 espécies)	https://bitbucket.org/biobakery/biobakery/wiki/metaphlan
Metaphyler	Liu *et al.* (2011)	31 marcadores filogenéticos	http:// metaphyler.cbcb.umd.edu
mOTU	Sunagawa *et al.* (2013)	10 genes marcadores de 3445 genomas procariotas de referência e espécies desconhecidas	http://www.bork.embl.de/*software*/mOTU/
PhyloSift	Darling *et al.* (2014)	37 famílias de genes	https://phylosift.wordpress.com/
TIPP	Nguyen *et al.* (2014)	31 marcadores filogenéticos	http://www.cs.utexas.edu/users/phylo/*software*/sepp/tipp-submission/

O *software Metaphyler* é um outro método de genes marcadores que, à semelhança do *MetaPhlAn*, usa o *BLAST* para o alinhamento das sequências e não inclui o gene *16S rRNA* na base de dados de sequências de referência (Liu *et al.*, 2011). O *Metaphyler* baseia a classificação taxonómica dos metagenomas com base em 31 marcadores filogenéticos de referência. Uma inovação deste método é o facto de permitir definir diferentes valores limite de classificação das sequências em função do comprimento dos *high-scoring pairs* obtidos no *BLAST*, do gene de referência em causa e do nível taxonómico em análise (Liu *et al.*, 2011).

Os métodos de genes marcadores podem também ser usados no âmbito da análise filogenética de comunidades de microorganismos. O *PhyloSift* é um programa que usa uma base de dados de sequências de 37 famílias de genes consideradas "universais" em termos dos genomas bacterianos e de cópia-única que representam cerca de 1% do genoma bacteriano médio (Darling *et al.*, 2014). Ao contrário do *MetaPhlAn* e do *Metaphyler*, as sequências homólogas são obtidas com o programa *LAST* e seguidamente alinhadas usando o algoritmo *hhmmalign* do pacote *HMMER* (Eddy, 2011) contra um perfil resultante de um alinhamento múltiplo de sequências de referência, gerado de acordo com um modelo oculto de Markov (MOM) aplicado a cada família de sequências génicas. As sequências alinhadas para cada uma das famílias de genes são concatenadas para produzir uma sequência única, a qual é colocada na árvore filogenética de referência usando o programa *pplacer* (Matsen *et al.*, 2010) de acordo com o método de máxima verosimilhança ou um método Bayesiano.

1.5. Métodos de montagem de leituras de sequenciação *shotgun*

Os métodos de montagem de leituras ("assembly") têm como finalidade reconstruir a sequência genómica de cada um dos genomas presentes numa comunidade de microorganismos, a partir das leituras dos diversos fragmentos de DNA que foram sequenciados. A reconstrução das sequências genómicas em metagenomas não é uma tarefa fácil por vários motivos. Uma das razões é que a cobertura das leituras de sequências pode ser muito diferente para diferentes genomas, uma vez que é

frequente existirem espécies com elevado predomínio na comunidade e outras com presença residual. Os programas de montagem de genomas individuais *de novo*, como o *Velvet* (Zerbino and Birney, 2008), estão preparados para reconstruir genomas com cobertura relativamente uniforme e, como tal, não estão indicados para análise de dados de metagenómica *shotgun*. Um outro motivo é que no caso de existirem 2 ou mais estirpes da mesma espécie na mesma amostra, cujas sequências genómicas são altamente similares, os métodos de montagem apresentam dificuldade em separar os respectivos *contigs*. Por fim, estes métodos são computacionalmente bastante exigentes, particularmente quando se analisa um número muito elevado de leituras, o que requer a disponibilidade de uma infraestrutura de *hardware* de elevada capacidade de computação.

Alguns dos programas que aplicam métodos de montagem para análise de metagenomas encontram-se descritos na **tabela 1.8**. A maioria dos programas baseia-se no método de grafos dirigidos de *de Bruijn* e na composição de *k-mers*, no qual a conectividade do grafo, em que cada nodo representa uma dada sequência de *k-mer*, permite reconstruir a sequência genómica seguindo um caminho ao longo do grafo dirigido de acordo com a sobreposição da sequência dos *k-mers* individuais presentes em nodos adjacentes (Zerbino and Birney, 2008). O *MetaVelvet* é um *software* de análise de leituras de sequenciação metagenómica *shotgun*, implementado a partir do *Velvet*, e que também se baseia na construção de grafos dirigidos de *de Bruijn* e na composição de *k-mers* (Namiki *et al.*, 2012). O princípio do *MetaVelvet* é o de que cada genoma que constitui parte dos genomas da amostra, tem o seu próprio grafo, o qual pode ser extraído através da decomposição do grafo geral em subgrafos de acordo com o número de genomas presentes em cada amostra (Namiki *et al.*, 2012).

O *Ray Meta* é outro *software* que utiliza a técnica de grafos de *de Bruijn*, mas que está assente numa estrutura informática distribuída e escalável de elevada capacidade computacional, a qual permite fazer a montagem de conjuntos de dados constituídos por 3 biliões de leituras (Boisvert *et al.*, 2012). A montagem dos genomas e a obtenção dos perfis taxonómicos são efectuados utilizando uma modificação do algoritmo do *software Ray* v2.0.0 (Boisvert *et al.*, 2010), que consiste em reconstruir os genomas através de um grafo de *de Bruijn* e depois aplicar a técnica de "grafos de *de Bruijn* coloridos" para derivar o perfil taxonómico da respectiva comunidade de microorganismos (Boisvert *et al.*, 2012). A técnica de coloração dos grafos consiste em atribuir a cada vértice uma cor distinta por cada sequência de referência. Por exemplo, se um *k-mer* tiver apenas 1 cor atribuída, então é porque pertence a apenas uma sequência de referência da base de dados. Isto permite atribuir um genoma a um *táxon* de baixo nível ou, se tal não for possível, utilizar então o MAC desse genoma.

Outros métodos foram desenvolvidos com o intuito de tornar a necessidade de memória do computador independente da profundidade da sequenciação. É o caso do *software Latent Strain Analysis* (*LSA*) que permite fazer a montagem de metagenomas após partição das leituras originais (Cleary *et al.*, 2015). O método do *LSA* baseia-se na identificação de padrões covariantes de *k-mers* num conjunto de amostras em simultâneo. Os vários *k-mers* de um dado genoma devem variar em proporção dentro de uma mesma amostra, o mesmo acontecendo para os *k-mers* dos restantes genomas presentes nessa amostra. Ao analisarem-se várias amostras em paralelo, é de esperar que para cada genoma existente numa dada amostra, a proporção dos vários *k-mers* desse genoma varie entre amostras, mas se mantenha semelhante na mesma amostra. O primeiro passo do *LSA* consiste na construção de uma matriz de abundância bidimensional em que as linhas correspondem às diferentes amostras e as colunas correspondem a ~2 biliões de *k-mers* com ~33 nucleótidos cada. A matriz é decomposta em vectores ("eigenvectors") que contêm a covariância da abundância dos *k-mers* nas várias amostras, a qual é representada sob a forma de "eigengenomes" que são usados para fazer o agrupamento dos *k-mers* e a partição das leituras (Cleary *et al.*, 2015). Esta partição tem

Tabela 1.8. Programas de análise de leituras de sequenciação metagenómica *shotgun* por métodos de montagem ("assembly").

Programa	Referência	Método	URL
DIME	Guo *et al.* (2015)	Construção de um grafo de "pesos"; agrupamento de leituras após partição do grafo; construção de *contigs* em cada agrupamento usando por exemplo o *Genovo*; combinação de *contigs* candidatos	http://grid.cs.gsu.edu/~xguo9/research/DIME.html
Genovo	Laserson *et al.* (2011)	Montagem de genomas com base num modelo probabilístico gerador de formação de leituras	http://cs.stanford.edu/group/genovo/
IDBA-UD	Peng *et al.* (2012)	Construção de um grafo de *de Bruijn* usando *thresholds* variáveis para filtragem de *contigs* erróneos, montagem "local" através da informação de leituras emparelhadas e re-alinhamento de leituras erróneas de regiões de elevada cobertura em *contigs* bem estabelecidos	http://i.cs.hku.hk/~alse/hkubrg/projects/idba_ud/
InteMAP	Lai *et al.* (2015)	Pré-processamento de leituras para correção de erros; montagem de genomas usando os programas IDBA-UD, SGA (Simpson and Durbin, 2012) e/ou CABOG (Miller *et al.*, 2008), e junção das montagens em função do nível de cobertura	http://cqb.pku.edu.cn/ZhuLab/InteMAP/index.html
LSA (a)	Cleary *et al.* (2015)	Construção de matriz de abundância de *k-mers* (n=33) em cada amostra; decomposição da matriz para gerar conjunto de *eigengenomes*; agrupamento de *k-mers* através dos *eigengenomes* e respectiva partição de leituras; montagem das partições individuais e alinhamento com base de dados de sequências de referência	https://github.com/brian-cleary/LatentStrainAnalysis
MEGAHIT	Li *et al.* (2016)	Utilização de grafos de *de Bruijn* sucintos (Bowe *et al.*, 2012) e de múltiplos tamanhos de *k-mers*	https://github.com/voutcn/megahit
MeGAMerge	Scholz *et al.* (2014)	Funde os *contigs* obtidos por diferentes programas para leituras de plataformas *Illumina* e *454*/*Roche* com leituras longas obtidas pela metodologia de sequenciação de *Sanger* (Sanger *et al.*, 1977) ou pela plataforma *PacBio*, para ultrapassar as especificidades e limitações dos diferentes programas	https://github.com/LANL-Bioinformatics/MeGAMerge
Meta-IDBA	Peng *et al.* (2011)	Construção de grafos de *de Bruijn* eliminando os ramos devidos a regiões comuns em diferentes espécies (para as isolar umas das outras) e efectuando um alinhamento múltiplo consenso (para representar os *contigs* das várias sub-espécies de uma mesma espécie)	http://i.cs.hku.hk/~alse/hkubrg/projects/metaidba/
MetAMOS	Treangen *et al.* (2013)	*Pipeline* modular e personalizável para análise metagenómica composta por ferramentas para análise de qualidade (e.g., *FastQC*), montagem de leituras (e.g., *Meta_IDBA*), mapeamento de leituras (e.g., *bowtie2*) e anotação (e.g., *BLAST*).	http://cbcb.umd.edu/*software*/metAMOS
MetaVelvet	Namiki *et al.* (2012)	Decompõe um grafo de *de Bruijn* composto pelas leituras misturadas das várias espécies em sub-grafos (usando os nodos quiméricos) e constrói *scaffolds* a partir de cada sub-grafo como se tratassem de genomas de espécies individuais	http://metavelvet.dna.bio.keio.ac.jp/
MetaVelvet-SL	Afiahayati *et al.* (2015)	Programa de montagem baseado no *MetaVelvet* que usa um algoritmo de *machine-learning* supervisionado para classificar os nodos quiméricos candidatos e calcula a abundância para cada sub-grafo individual, de forma a identificar os nodos únicos de cada espécie	http://metavelvet.dna.bio.keio.ac.jp/MSL.html
MOCAT2	Kultima *et al.* (2016)	*Pipeline* composta por filtragem de qualidade de leituras, montagem de leituras, previsão e anotação funcional de genes, e determinação do perfil funcional da comunidade	http://mocat.embl.de/
Omega	Haider *et al.* (2014)	Baseado na utilização de grafos de sobreposição (Simpson and Durbin, 2012) e pretendendo ultrapassar as limitações dos programas de montagem de genomas individuais quando aplicados a amostras de metagenómica (e.g., resolver regiões de repetições existentes em 2 genomas através do cálculo da respectiva cobertura)	http://omega.omicsbio.org/
Ray Meta	Boisvert *et al.* (2012)	Método de montagem de leituras (baseado em grafos de *de Bruijn* coloridos) e caracterização do perfil funcional (usando a base de dados de taxonomia *Greengenes*), altamente escalável, distríbuido e de elevada performance, permitindo a análise de 3 biliões de leituras correspondentes a 1000 genomas diferentes	http://denovoassembler.sourceforge.net/
SAT-Assembler	Zhang *et al.* (2014)	Baseado no alinhamento inicial das leituras usando perfis de MOM de famílias génicas de interesse, seguido de montagem das leituras de cada família usando grafos de sobreposição	https://sourceforge.net/projects/sat-assembler/
SFA-SPA	Yang *et al.* (2015)	Algoritmo de montagem de leituras que utiliza um *suffix array*, que permite estender o caminho ao longo do grafo de *de Bruijn* usando prefixos ou sufixos sobreponíveis das leituras	https://sourceforge.net/projects/spa-assembler/

(a) Permite também a análise de genes marcadores.

como efeito reduzir a carga computacional que seria necessária para analisar todas as leituras em simultâneo. Assim, a partir deste ponto, pode efectuar-se a montagem das leituras e/ou recorrer a análise de genes marcadores, em cada partição, para fazer a identificação dos genomas presentes em cada amostra.

1.6. Métodos de combinação de cobertura e composição

Os métodos de combinação de cobertura e composição (designação atribuída neste trabalho sem equivalência directa em Inglês), representam uma abordagem mais recente para a análise de dados de sequenciação *shotgun* e que é substancialmente diferente dos métodos de classificação e de montagem atrás descritos. Como o próprio nome indica, os métodos de combinação de cobertura e composição usam a informação combinada da cobertura das sequências genómicas e da respectiva frequência de *k-mers*, em 2 ou mais amostras de metagenomas em simultâneo, para determinar as respectivas estruturas composicionais. As amostras respeitantes a cada conjunto de dados têm de estar espacial ou temporalmente relacionadas, podendo representar, por exemplo, diferentes locais de amostragem ou tempos distintos de recolha de um mesmo local. A **tabela 1.9** apresenta alguns dos métodos de combinação de cobertura e composição, disponíveis no domínio público, para análise de dados de sequenciação *shotgun*.

Tabela 1.9. Programas de análise de leituras de sequenciação metagenómica *shotgun* por métodos de combinação de cobertura e composição.

Programa	Referência	Método	URL
CONCOCT	Alneberg *et al.* (2014)	Efectua montagem de leituras em *contigs* com o program *Velvet* e mapeia as leituras nos *contigs* usando o *bowtie2*. Os fragmentos genómicos dos *contigs* das várias amostras são agrupados usando os dados de cobertura, as leituras emparelhadas e a respectiva frequência de *k-mers* (k=4)	https://github.com/BinPro/CONCOCT
GroopM	Imelfort *et al.* (2014)	Constroí *contigs* a partir de 3 ou mais bibliotecas usando por exemplo o *Velvet*, *IDBA-UD*, *Ray Meta* ou *SPAdes* (Bankevich *et al.*, 2012) e mapeia as leituras contra os *contigs* usando o *BWA*. O *binning* dos *contigs* é efectuado usando tendo por base a frequência de *k-mers* (k=4). A combinação destes *bins* com os ficheiros de mapeamento permite obter os *bins* genómicos, cuja integridade é calculada através da presença de genes marcadores de cópia única	http://ecogenomics.github.io/GroopM/
MetaBAT	Kang *et al.* (2015)	Recebe como entrada um ficheiro *.fasta* de montagem de metagenomas de múltiplas amostras que é usado para mapear as leituras. Para cada par de *contigs* é calculada a probabilidade da distância entre si usando a cobertura média por base e a frequência de tetranucleótidos de cada *contig*. Os genomas são alocados a *bins* usando um algoritmo de agrupamento *k-medoid* modificado, com base na matriz de distâncias compostas emparelhadas	https://bitbucket.org/berkeleylab/metabat
MyCC	Lin and Liao (2016)	Utiliza *contigs* de leituras como dados de entrada (produzidos por exemplo pelo *software Ray Meta*) a partir dos quais são extraídas as sequências de 40 genes marcadores. Para cada *contig* é determinada a frequência dos respectivos *k-mer* (k=4), a partir dos quais são gerados agrupamentos, os quais são corrigidos em função das sequências dos genes marcadores	https://sourceforge.net/projects/sb2nhri/files/MyCC/

Os métodos de combinação de cobertura e composição recebem como dados de entrada os *contigs* resultantes da montagem das sequências genómicas de cada amostra ou incorporam no próprio método a montagem prévia das leituras usando por exemplo os programas *Velvet*, *IDBA-UD* ou *Ray Meta*

(Bankevich *et al.*, 2012). Estes métodos englobam-se na categoria dos métodos não supervisionados, ou seja, que não necessitam do conhecimento prévio de sequências de referência. A ideia que está na base destes métodos é a de que os *contigs* que apresentam perfis de cobertura idênticos deverão ter sido originados a partir da mesma população de microorganismos. Deste modo, a combinação dos dados de cobertura com a informação da composição da sequência, através do cálculo da frequência de *k-mers* (com *k* frequentemente igual a 4), pode melhorar o *binning* de metagenomas relativamente a outros métodos (Imelfort *et al.*, 2014). Os perfis de cobertura podem ser determinados através do mapeamento das leituras contra parte ou a totalidade dos respectivos *contigs* produzidos com base nessas mesmas leituras (Imelfort *et al.*, 2014). A integridade dos *bins* gerados pelos métodos de combinação de cobertura e composição pode ser avaliada através da análise de genes marcadores de cópia única (Imelfort *et al.*, 2014).

1.7. Análise de leituras de sequenciação do gene *16S rRNA*

1.7.1. Bases de dados de sequências de genes ribossomais

A relevância da análise do DNA ribossomal, quer como instrumento de investigação na identificação de novas espécies ou na caracterização de comunidades de microorganismos, quer como técnica laboratorial aplicada no âmbito clínico, conduziu inevitavelmente à criação de várias colecções organizadas de sequências dos genes ribossomais (**tabela 1.10**). De acordo com a *RNACentral* (RNAcentral: The non-coding RNA sequence database, 2015), que importa sequências dos genes ribossomais de diferentes bases de dados, estão atualmente disponíveis pelo menos 25 bases de dados destas sequências. Uma das primeiras bases de dados de genes ribossomais, a *European Ribossomal RNA Database* (*ERRD*), nasceu na Universidade de Antuérpia em 1983 (Erdmann *et al.*, 1983), tendo sofrido diversas actualizações até 2007. A base de dados foi constituída por sequências alinhadas das subunidades pequena e grande do RNA ribossomal, extraídas de sequências existentes na base de dados do *European Molecular Biology Laboratory* (De Rijk *et al.*, 2000; Van de Peer *et al.*, 2000). Após alinhamento das sequências, estas eram comparadas através de métodos de super-imposição das estruturas secundárias, e corrigidas se necessário (uma vez que a estrutura proteica deverá ser mais conservada que a sequência de DNA), podendo eventualmente obrigar a correcções também ao nível do alinhamento das estruturas primárias (Wuyts *et al.*, 2004). O surgimento e crescimento da base de dados *SILVA* (ver abaixo) ditou o fim da *ERRD*, embora os dados estejam ainda disponíveis (European Ribossomal RNA Database, 2002).

A base de dados *16S Ribossomal RNA Mutation Database* (*16SMDB*), também conhecida como *Triman Database,* foi criada por Kathleen Triman do *Franklin and Marshall College* da Pensilvânia no início dos anos 90, que continua a ser a sua curadora na actualidade. A *16SMDB* consiste numa listagem das mutações encontradas no gene *16S rRNA* de *E. coli*, acrescida do respectivo fenótipo mutante, da indicação se a mutação foi observada *in vivo* ou *in vitro*, e das respectivas referências bibliográficas (Triman, 1994). As actualizações subsequentes desta base de dados foram publicadas em 1996 (Triman, 1996) e 1997 (Triman and Adams, 1997), incluindo-se também nesta última versão uma lista anotada das mutações do gene *23S rRNA* de *E. coli*. Em 1998 foram publicadas versões expandidas das bases de dados das mutações nos genes *16S rRNA* e *23S rRNA* encontradas em outros microorganismos, descritas de acordo com o sistema de numeração de *E. coli* (Triman *et al.*, 1998). Em 2007, foi ainda publicada uma revisão das mutações de proteínas ribossomais, divididas por

Tabela 1.10. Resumo das principais bases de dados de sequências do gene *16S rRNA*.

Base de Dados	Tipo de dados	Organismos	URL	Nº de registos (em 29/07/2017)	Referência
16S ribossomal RNA Mutation Database	Sequências, estruturas e mutações dos genes ribossomais, incluindo fenótipo, entre outros	*E. coli* (maioria) e outros organismos	http://www.rna.ccbb.utexas.edu/SAE/2B/triman.php	42143 (*16S rRNA*), 985 (*5S rRNA*), 5275 (*23S rRNA*) sequências	Triman (1998)
CORE	Sequências do gene *16S rRNA* obtidas do microbioma oral de amostras clínicas	*Bacteria*	http://microbiome.osu.edu/home	> 1000	Griffen *et al.* (2011)
EzTaxon/EzTaxon-e/EzBioCloud	Sequências do gene *16S rRNA* e de genomas de procariotas	*Bacteria* e *Archaea*	http://www.ezbiocloud.net/dashboard	62685 (sequências *16S rRNA*) e 82607 (genomas)	Chun *et al.* (2007), Kim *et al.* (2012), Yoon *et al.* (2017)
Greengenes	Sequências do gene *16S rRNA*	*Bacteria* e *Archaea*	http://greengenes.lbl.gov/Download/	1012869 (sequências *16S rRNA*)	DeSantis *et al.* (2006a), McDonald *et al.* (2012)
HOMD	Sequências do gene *16S rRNA* e de genomas de procariotas do microbioma oral e respectivos metadados	*Bacteria*	http://www.homd.org/	1362 (genomas) e 688 (táxones)	Chen *et al.* (2010)
Meta-Storms/GPU-Meta-Storms	Amostras de estudos de metagenómica e respectivas anotações	*Bacteria* e *Archaea*	http://www.computationalbioenergy.org/meta-storms.html	Não indicado	Su *et al.* (2012b), Su *et al.* (2014b)
probeBase	Sequências de oligonucleótidos e sondas para os genes *16S rRNA* e *23S rRNA*	*Bacteria*, *Archaea* e *Eukarya*	http://www.probebase.net	Não indicado	Greuter *et al.* (2016)
Rfam	Sequências múltiplas alinhadas, estruturas secundárias consenso e modelos de covariância de 2687 famílias de RNAs	*Bacteria*, *Archaea* e *Eukarya*	http://rfam.xfam.org/	265292 (*Eukarya*), 165776 (*Bacteria*) e 25385 (*Archaea*) sequências alinhadas da sub-unidade	Nawrocki *et al.* (2015)
Ribossomal Database Project	Sequências dos genes *16S rRNA* e *18S rRNA*, e da subunidade pequena das mitocôndrias	*Bacteria*, *Archaea* e *Eukarya* (fungos)	http://rdp.cme.msu.edu/index.jsp	3356809 (*16S rRNA*) e 125525 (*28S rRNA* de fungos) sequências	Cole *et al.* (2014)
rrnDB	Número de cópias dos genes *16S rRNA* e *23S rRNA*	*Bacteria* e *Archaea*	https://rrndb.umms.med.umich.edu/	2642 (*Bacteria*) e 188 (*Archaea*) espécies	Stoddard *et al.* (2015)
SILVA	Sequências alinhadas dos genes *16S rRNA*/*18S rRNA* e *23S rRNA*/*28S rRNA*	*Bacteria*, *Archaea* e *Eukarya*	http://www.arb-silva.de/	5616941 (*16S rRNA*/*18S rRNA*) e 735238 (*23S rRNA*/*28S rRNA*) sequências	Quast *et al.* (2013)

subunidade ribossomal, e mutações de factores ribossomais, divididas por função ribossomal (Triman, 2007).

Actualmente, três bases de dados são responsáveis por armazenar um grande volume de sequências. As bases de dados *Ribossomal Database Project* (*RDP*), *Greengenes* e *SILVA*, contêm cerca de 1,3M, 3,2M e 5,4M de sequências do gene *16S rRNA*, respectivamente. A base de dados *RDP* foi lançada no início de 1992, por iniciativa de Carl Woese e Gary Olsen, com os objectivos de fornecer à comunidade científica uma base de dados de sequências do gene *16S rRNA* (Olsen *et al.*, 1991). Este projecto tem sofrido, ao longo dos anos, múltiplas actualizações de dados e incorporação de novas ferramentas de análise (Larsen *et al.*, 1993; Maidak *et al.*, 1994; Maidak *et al.*, 1996; Maidak *et al.*, 1997; Maidak *et al.*, 1999; Maidak *et al.*, 2000; Maidak *et al.*, 2001; Cole *et al.*, 2003; Cole *et al.*, 2005; Cole *et al.*, 2007; Cole *et al.*, 2009; Cole *et al.*, 2014). O *RDP* fornece também diversos serviços de análise de dados na *internet*, incluindo o programa de classificação de sequências *Ribossomal Database Project* (*RDP*) *Classifier* (Wang *et al.*, 2007). A *Greengenes* é uma base de dados de sequências do gene *16S rRNA*, à qual está associado um conjunto vasto de ferramentas para análise e tratamento de dados e que inclui a possibilidade de efectuar alinhamento de sequências, procurar sequências de oligonucleótidos ou sondas, efectuar análise filogenética e fazer o pré-processamento de leituras usando valores de qualidade (DeSantis *et al.*, 2006a). A base de dados *SILVA* contém sequências alinhadas dos genes que codificam para as subunidades do RNA ribossomal 16S/18S e 23S/28S, para todos os domínios da vida (Quast *et al.*, 2013, Yilmaz *et al.*, 2014). Além do conjunto total de sequências destes genes, a *SILVA* disponibiliza também bases de dados de sequências previamente submetidas a controlo de qualidade (designadas como referência), cujo comprimento mínimo é bastante superior ao das sequências que se encontram nas bases de dados gerais.

O elevado número de sequências disponíveis do gene *16S rRNA* obriga a esforços no sentido de eliminar a redundância e fornecer sequências identificadoras de estirpes de microorganismos. Por exemplo, a base de dados *SILVA* contém actualmente 645.151 sequências não redundantes da subunidade pequena do RNA ribossomal. Uma outra base de dados, designada *EzTaxon*, constitui uma colecção curada de sequências do gene *16S rRNA* dos procariotas, e foi inicialmente estabelecida com o propósito de disponibilizar sequências validadas de estirpes-tipo (*type strains*) para identificação taxonómica de novos isolados (Chun *et al.*, 2007). Mais tarde, esta base de dados foi estendida para dar origem à base de dados *EzTaxon-e*, a qual incorpora sequências do gene *16S rRNA* de organismos procariotas não cultiváveis, obtidas a partir de outras bases de dados públicas ou extraídas de sequências de genomas ou metagenomas previamente montados (Kim *et al.*, 2012). A *EzTaxon-e* disponibiliza também uma filogenia completa de todas as estirpes-tipo e filotipos (ou seja, espécies putativas de procariotas existentes na natureza que têm uma sequência única do gene *16S rRNA*) existentes na base de dados (Kim *et al.*, 2012).

As bases de dados atrás referidas podem considerar-se de tipo generalista, ou seja, procuram coleccionar todas as sequências de um ou mais genes ribossomais independentemente do bioma em que foram obtidas. No entanto, o crescente interesse na caracterização do microbioma humano para fins clínicos, conduziu à criação de bases de dados específicas para este fim. Por exemplo, a base de dados *CORE* tem como objectivo criar uma colecção de sequências do gene *16S rRNA* representativas do microbioma oral "nuclear" (Griffen *et al.*, 2011). A *CORE* foi construída usando sequências derivadas de estudos do microbioma humano oral e sequências obtidas através da sequenciação do gene *16S rRNA* em amostras clínicas. Esta base de dados disponibiliza uma árvore filogenética das sequências do gene *16S rRNA* do microbioma oral, construída com base no método da máxima verosimilhança. À semelhança da *CORE*, a *Human Oral Microbiome Database* (*HOMD*) é também

uma base de dados de sequências curadas do gene *16S rRNA*, pertencentes a cerca de 400 táxones de procariotas que habitam a cavidade oral humana (Chen *et al.*, 2010; Dewhirst *et al.*, 2010). Uma mais-valia da *HOMD* é a de integrar dados fenotípicos, filogenéticos, genómicos, clínicos e bibliográficos através de um número identificador único (Chen *et al.*, 2010).

Nos estudos de sequenciação metagenómica que usam o gene *16S rRNA* como marcador, a variação do número de cópias deve ser considerada nos cálculos da abundância relativa de cada estirpe, a fim de evitar o enviesamento causado pela contagem directa das leituras de sequenciação. A base de dados *Ribosomal RNA Operon Copy Number Database* (*rrnDB*) é um forte auxiliar neste sentido, uma vez que tem como objectivo catalogar o número de cópias por genoma dos genes do RNA ribossomal 5S, 16S e 23S, entre outros genes, dos microorganismos pertencentes aos domínios *Bacteria* e *Archaea* (Klappenbach *et al.*, 2001; Lee *et al.*, 2009; Stoddard *et al.*, 2015). A *rrnDB* é construída usando como fontes de dados o *Kyoto Encyclopedia of Genes and Genomes* (KEGG, Kanehisa and Goto, 2000), o *NCBI*, o *RDP Classifier* e os resultados obtidos a partir de diferentes métodos empíricos de determinação do número de cópias do gene *16S rRNA*.

Além das sequências de DNA, as estruturas secundárias das moléculas do RNA ribossomal podem ser importantes na identificação de sequências homólogas, uma vez que a estrutura está frequentemente melhor conservada do que a sequência. A *Rfam* é uma base de dados de famílias de RNAs que contém RNAs não codificantes, elementos estruturados de regulação em *cis* e RNAs de *self-splicing* (Nawrocki *et al.*, 2015). Cada família está representada por um alinhamento múltiplo que pode também incluir a anotação da estrutura secundária consenso e um modelo de covariância construído usando o *software Infernal* (Nawrocki and Eddy, 2013). Os modelos criados neste *software* podem ser usados para localizar sequências de RNA homólogas em bases de dados, podendo subsequentemente conduzir a alinhamentos múltiplos de maior dimensão.

A comparação de metagenomas em larga escala é uma das áreas que tem despertado o interesse dos investigadores que se dedicam aos estudos de metagenómica. Este propósito obriga a pesquisar repositórios de amostras para verificar se existem semelhanças com as amostras desconhecidas, e quantificar essa mesma semelhança (Su *et al.*, 2012b). Neste sentido, foi criado o *Meta-Storms* que, além de disponibilizar uma base de dados de amostras de estudos de metagenómica (que contém conjuntos de dados organizados e indexados de metagenomas), oferece também a possibilidade de pesquisar uma amostra contra a base de dados usando uma função baseada em filogenia quantitativa (Su *et al.*, 2012b). Esta quantificação pode ser baseada na sobreposição entre táxones pertencentes a diferentes comunidades de microorganismos ou na sobreposição das árvores filogenéticas dessas mesmas comunidades. Mais recentemente, foi lançada uma nova versão, designada *GPU-Meta-Storms* (Su *et al.*, 2014b), com o objectivo de aumentar a capacidade computacional associada à quantificação da semelhança filogenética entre amostras.

As bases de dados de sequências dos genes ribossomais também podem ser úteis no desenho de oligonucleótidos para PCR, uma vez que a elevada semelhança destas sequências entre diferentes espécies obriga a uma selecção criteriosa dos oligonucleótidos óptimos para cada estudo (Klindworth *et al.*, 2013). A *probeBase* é uma base de dados constituída por sequências de oligonucleótidos de PCR e de sondas de *fluorescence in situ hybridization*, para os genes *16S rRNA* e *23S rRNA* (Greuter *et al.*, 2016). A base de dados permite procurar oligonucleótidos ou sondas para organismos específicos, ou para sequências submetidas pelo utilizador, e verificar a especificidade e cobertura das sequências nas bases de dados *SILVA* e *RDP*, assim como a eficiência de hibridação *in silico* dos oligonucleótidos ou sondas.

1.7.2. Classificação taxonómica de leituras de amplicões do gene *16S rRNA*

Os métodos de classificação taxonómica dos amplicões do gene *16S rRNA* são essencialmente de 2 tipos: métodos de homologia (*homology-based*) e de agrupamento (*prediction-based*). A **figura 1.4** apresenta um diagrama dos principais níveis de análise secundária de dados de sequenciação de amplicões do gene *16S rRNA*, incluindo alguns dos programas disponíveis. Os métodos de homologia podem ser baseados na análise de semelhança, na composição de *k-mers* ou em MOM que permitem classificar sequências parciais ou completas do gene *16S rRNA* (Ghosh *et al.*, 2012). Uma das desvantagens dos métodos de homologia é o facto de, na maioria dos programas disponíveis, as leituras serem comparadas directamente uma a uma às respectivas sequências de referência, o que tem consequências importantes em termos do tempo total de computação. Pelo contrário, na abordagem de agrupamento, procura reduzir-se o conjunto de leituras a analisar ao agruparem-se estas em UTO(s), de acordo com a distância genética ou com métodos probabilísticos (Chatterjee *et al.*, 2014). Apenas as sequências representativas (*consensus*) de cada UTO são usadas para a classificação taxonómica e construção da árvore filogenética, enquanto o número total de sequências em cada UTO representa a abundância do respectivo grupo taxonómico (Tanaseichuk *et al.*, 2014).

A **tabela 1.11** apresenta um resumo de alguns dos métodos de classificação taxonómica de amplicões do gene *16S rRNA* existentes no domínio público, incluindo 8 métodos de homologia e 10 métodos de agrupamento. O programa *Quantitative Insights Into Microbial Ecology* (*QIIME*, que se pronuncia "chime") é um pacote completo de ferramentas bioinformáticas que inclui desde a desmultiplicação das leituras geradas pelas plataformas de sequenciação, até à interpretação dos resultados e deposição das sequências em bases de dados (Caporaso *et al.*, 2010a). Uma das mais-valias do *QIIME* é o de permitir escolher diferentes abordagens para as várias etapas, como seja a selecção de diferentes algoritmos para classificação taxonómica por homologia (usando o *BLAST* ou o *RDP Classifier*, por exemplo) ou a utilização de diferentes métodos filogenéticos, o que traz a vantagem óbvia de permitir comparações entre resultados (Caporaso *et al.*, 2010a).

O *RDP Classifier* permite atribuir as sequências do gene *16S rRNA* a grupos taxonómicos que vão desde o domínio até ao género. O algoritmo utiliza todas as "palavras" (*k-mers*) possíveis de 8 bases (conjunto *W*) como base para a classificação das sequências e pode ser decomposto nas seguintes fases (Wang *et al.*, 2007):

i) Calcular a probabilidade de encontrar a palavra w_i numa sequência do gene *16S rRNA*, sendo $n(w_i)$ o número de sequências que contêm a palavra w_i: $P_i = [n(w_i) + 0.5]/(N+1)$, em que *N* representa o número total de sequências;

ii) Calcular a probabilidade condicional que um membro de um género *G*, com um conjunto de teste composto por *M* sequências, contém a palavra w_i: $P(w_i|G) = [m(w_i) + Pi]/(M+1)$, em que $m(w_i)$ representa o número de sequências de *M* que contêm a palavra w_i;

iii) Calcular a probabilidade conjunta de se observar uma sequência parcial (*S*) pertencente ao género *G*, contendo um conjunto de palavras *V*: $P(S|G) = \prod P(v_i|G)$

iv) De acordo com o teorema de Bayes, a probabilidade de que uma sequência desconhecida *S* pertence ao género *G* é dada por: $P(G|S) = P(S|G)xP(G)/P(S)$. Uma vez que é teoricamente possível encontrar qualquer género e, consequentemente, qualquer sequência numa amostra, a probabilidade de uma sequência pertencer a um determinado género pode ser estimada apenas por $P(S|G)$, sendo que cada sequência é atribuída a um dado grupo taxonómico de acordo com o valor mais elevado de probabilidade (Wang *et al.*, 2007).

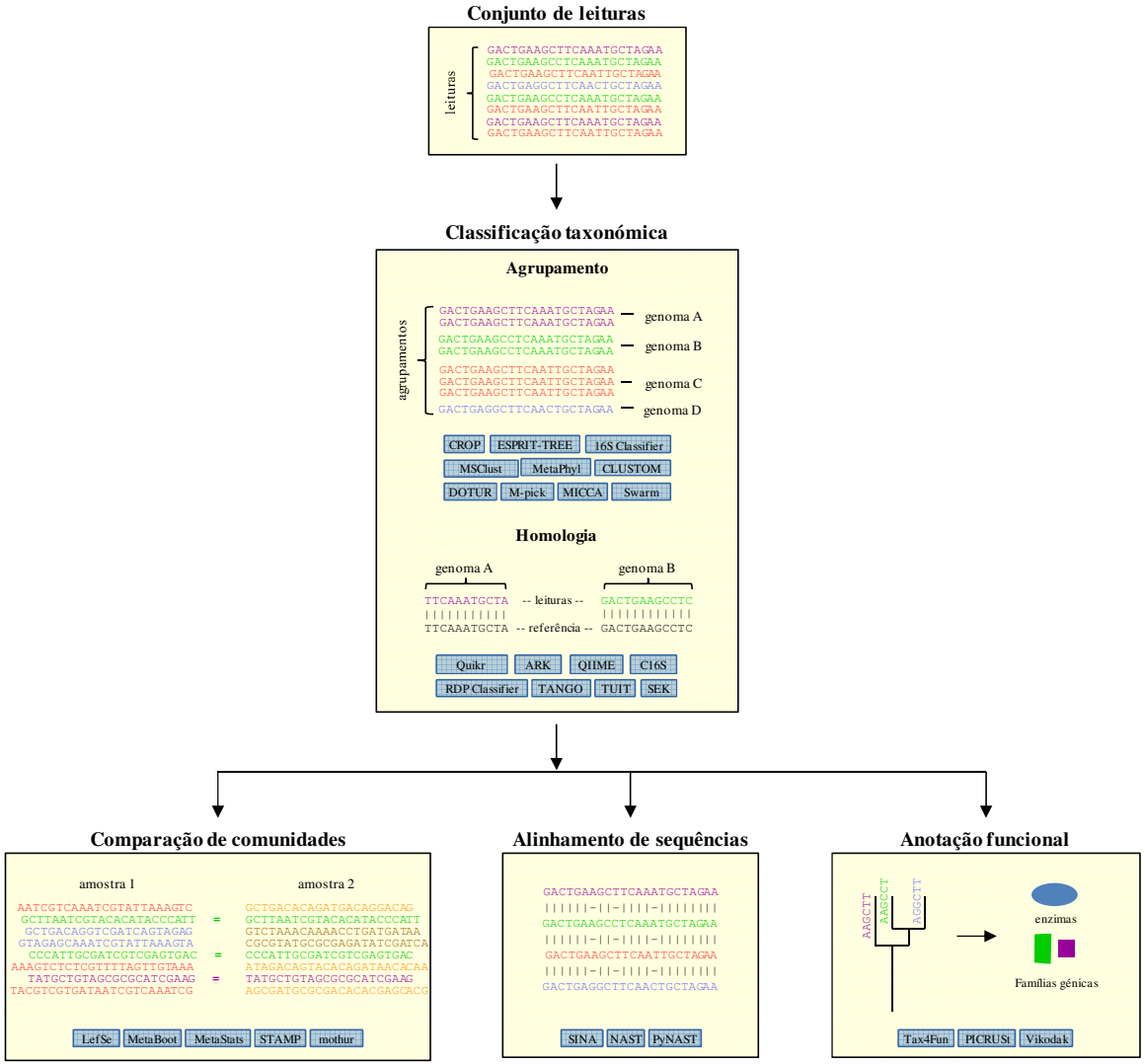

Figura 1.4. Diagrama representativo dos vários níveis da análise secundária de dados de sequenciação do gene *16S rRNA*. Os objectivos desta análise passam por determinar a composição da comunidade de microorganismos, conhecer o seu perfil funcional, construir alinhamentos de sequências do gene *16S rRNA* e comparar as características da comunidade com comunidades amostradas em outros locais ou em alturas difererentes. Os programas usados em cada nível de análise estão indicados nas caixas azuis.

Tabela 1.11. Programas de análise de leituras de sequenciação do gene *16S rRNA*.

Programa	Referência	Tipo de método/descrição	URL
16S Classifier (a)	Chaudhary *et al.* (2015)	Agrupamento/algoritmo de *machine-learning* ("random forest")	http://metagenomics.iiserb.ac.in/16Sclassifier/application.php
ARK	Koslicki *et al.* (2015)	Homologia/pré-agrupamento das leituras com base em *k-mers* (algoritmo *K-means*) seguido da computação da composição por método de homologia usando os programas *Taxy* (Meinicke *et al.*, 2011), *Quikr* (Koslicki *et al.*, 2013) ou *SEK* (Chatterjee *et al.*, 2014)	https://github.com/dkoslicki/ARK
C16S (b)	Ghosh *et al.* (2012)	Homologia/classifica as leituras de acordo com MOM específicos de género e atribui cada leitura ao nível taxonómico mais elevado, de acordo com a qualidade do alinhamento	http://metagenomics.atc.tcs.com/C16S/
CLUSTOM (a)	Hwang *et al.* (2013)	Agrupamento/baseado na seleção de sequências "nucleares" localizadas no centro das distribuições representadas pelas distâncias genéticas das sequências	http://clustom.kribb.re.kr
CROP	Hao *et al.* (2011)	Agrupamento/baseado num método de agrupamento Bayesiano não supervisionado	https://github.com/tingchenlab/CROP
DOTUR	Schloss and Handelsman (2005)	Agrupamento/baseado em algoritmos de agrupamento *nearest neighbor*, *furthest neighbor*, e *average neighbor*	https://github.com/*mothur*/DOTUR
ESPRIT-TREE	Cai and Sun (2011)	Agrupamento/baseado num agrupamento hierárquico de leituras através de um algoritmo rápido de pesquisa do *closest-pair*	http://plaza.ufl.edu/sunyijun/ES-Tree.htm
MetaPhyl	Tanaseichuk *et al.* (2014)	Agrupamento/utiliza as semelhanças entre UTO(s) de acordo com uma árvore filogenética	http://alumni.cs.ucr.edu/~tanaseio/metaphyl.htm
MICCA	Albanese *et al.* (2015)	Agrupamento/*pipeline* de análise metagenómica que inclui filtragem de qualidade das leituras, agrupamento das leituras em UTO(s), classificação taxonómica e inferência da árvore filogenética	http://micca.org/
M-pick	Wang *et al.* (2013)	Agrupamento/constrói um grafo de pesos em que os vértices constituem as leituras e as arestas representam a semelhança (“peso”) entre cada 2 leituras	http://plaza.ufl.edu/xywang/Mpick.htm
MSClust	Chen *et al.* (2013)	Agrupamento/método heurístico baseado em *multi-seeds* que reduz a necessidade de memória para agrupar as leituras	http://bioinformatics.dreamhosters.com/?page_id=113#Genomic_Protein_Sequence_Analysis
QIIME	Caporaso *et al.* (2010a)	Homologia/baseado por ex. no *BLAST* ou agrupamento/baseado por ex. no *UCLUST* (Edgar, 2010)	http://qiime.org/
Quikr	Koslicki *et al.* (2013)	Homologia/baseado na frequência de *k-mers* (k=6) e no método matemático de *compressive sensing*	https://sourceforge.net/projects/quikr/
RDP Classifier (a)	Wang *et al.* (2007)	Homologia/ baseado na frequência de *k-mers* (k=8) e no teorema de Bayes	http://rdp.cme.msu.edu/classifier/classifier.jsp
SEK	Chatterjee *et al.* (2014)	Homologia/baseado na frequência de *k-mers* (sendo *k* definido pelo utilizador) e na resolução de um sistema de equações lineares	https://github.com/dkoslicki/SEK
Swarm	Mahé *et al.* (2015)	Agrupamento/baseado na aglomeração de amplicões similares para gerar um conjunto inicial de UTO(s) e na utilização da abundância destes amplicões para subdivir as UTO(s)	https://github.com/torognes/swarm
TANGO	Alonso-Alemany *et al.* (2014)	Homologia/baseado no *BLAST* para alinhamento e no método do menor ancestral comum para a atribuição taxonómica	https://sourceforge.net/projects/taxoassignement/
TUIT	Tuzhikov *et al.* (2014)	Homologia/baseado no *BLAST* e pode ser complementado com o *RDP Classifie*r para melhorar os resultados da classificação taxonómica	https://sourceforge.net/projects/tuit/

(a) Estes programas também podem ser utilizados como um serviço *web*; (b) O *C16S* apenas pode ser utilizado como um serviço *web*.

1.7.3. Detecção de sequências do gene *16S rRNA* a partir de leituras de sequenciação *shotgun*

A sequenciação *shotgun* de metagenomas produz em teoria leituras de todos os genes do genoma, incluindo os das subunidades ribossomais como o gene *16S rRNA*. Neste sentido, as leituras que contenham sequências deste gene podem ser usadas para a classificação taxonómica como se se tratasse de uma sequenciação de amplicões do gene *16S rRNA*. Desta forma, reduzem-se significativamente os recursos computacionais e o tempo necessários para classificar taxonomicamente o conjunto de leituras produzidas pela sequenciação *shotgun*. No domínio público estão disponíveis pelo menos 5 programas que permitem detectar e/ou extrair sequências do gene *16S rRNA* a partir de conjuntos de leituras de sequenciação *shotgun* (**tabela 1.12**). Dado que as sequências do gene *16S rRNA* são relativamente conservadas nas bactérias e arqueias, é possível localizá-las no conjunto de dados, por exemplo, através de MOM (Lagesen *et al.*, 2007, Huang *et al.*, 2009).

Tabela 1.12. Programas de detecção de sequências do gene *16S rRNA* a partir de leituras de sequenciação *shotgun*.

Programa	Referência	Descrição	URL
i-rDNA (a)	Mohammed *et al.* (2011b)	Detecção de leituras do gene *16S rRNA* a partir de agrupamentos de sequências gerados com base em padrões de composição de tetranucleótidos	http://metagenomics.atc.tcs.com/i-rDNA/
Megraft	Bengtsson *et al.* (2012)	Extracção das leituras de fragmentos dos genes das subunidades pequenas (16S e 18S) do ribossoma para construção das sequências completas dos genes ribossomais	http://microbiology.se/software/megraft/
meta-rna (b)	Huang *et al.* (2009)	Detecção de leituras dos genes das subunidades ribossomais 5S, 16S e 23S dos procariotas usando MOM	http://weizhong-lab.ucsd.edu/meta_rna/
Metaxa2	Bengtsson-Palme *et al.* (2015)	Extracção de leituras parciais dos genes das subunidades ribosomais (16S, 18S, 23S e 28S) com classificação até ao nível de género ou espécie	http://microbiology.se/software/metaxa2/
RNAmmer	Lagesen *et al.* (2007)	Anotação de genes ribossomais de procariotas e eucariotas a partir de MOM	http://www.cbs.dtu.dk/services/RNAmmer/

(a) O *i-rDNA* pode ser acedido através de um serviço *web* em http://metagenomics.atc.tcs.com/i-rDNA/; (b) o *meta-rna* pode ser acedido através de um serviço *web* em http://weizhong-lab.ucsd.edu/metagenomic-analysis/server/hmm_rRNA/.

O *RNAmmer* (Lagesen *et al.*, 2007) é o programa mais usado na literatura para anotação de genes ribossomais em conjuntos de dados de sequenciação *shotgun*. Este programa permite detectar sequências dos genes das subunidades ribossomais 5S, 16S e 23S em procariotas, e dos genes das subunidades ribossomais 5S, 18S e 28S em eucariotas, a partir de MOM treinados a partir de alinhamentos estruturais múltiplos. O *meta-rna* também utiliza MOM para cada um dos genes das subunidades ribossomais dos procariotas, que foram construídos com base em alinhamentos múltiplos derivados de informação estrutural e de alinhamento de sequências, obtida a partir de diferentes bases de dados, permitindo identificar sequências dos genes ribossomais mesmo que incompletas (Huang *et al.*, 2009). A desvantagem destes 2 programas é que requer que cada uma das sequências do conjunto de dados seja analisada por cada modelo, o que tem implicações importantes em termos do tempo de computação, embora no caso do *RNAmmer* os autores tenham desenvolvido modelos alternativos que analisam apenas uma pequena porção (75 pb) de cada sequência (Lagesen *et al.*, 2007).

De acordo com Mohammed *et al.* (2011b), estimou-se que a percentagem de sequências pertencentes ao gene *16S rRNA*, no conjunto total de sequências obtidas por sequenciação *shotgun*, é normalmente inferior a 0,2%. O programa *identification of ribossomal DNA* (*i-rDNA*) foi desenvolvido para ser utilizado como uma ferramenta prévia ao *meta-rna*, com o objectivo de reduzir o tempo necessário à identificação de sequências do gene *16S rRNA* em conjuntos de dados de grande

dimensão. O *i-rDNA* usa as frequências de tetranucleótidos do conjunto total de genomas de bactérias e arqueias conhecidos, por forma a gerar agrupamentos de sequências com base em padrões similares de tetranucleótidos, sendo de esperar que alguns agrupamentos estejam enriquecidos em sequências do gene *16S rRNA* uma vez que este gene é conservado evolutivamente naqueles microorganismos. Em simultâneo, o algoritmo identifica os agrupamentos que contêm sequências com composição idêntica a cada uma das leituras pesquisadas. Assim, uma sequência é classificada como "provável fragmento do gene *16S rRNA*" se a percentagem de sobreposição entre o conjunto de agrupamentos previamente classificados como prováveis agrupamentos de sequências do gen*e 16S rRNA*, e os agrupamentos contendo sequências semelhantes à sequência pesquisada, exceder um limite inicialmente fixado pelo utilizador (Mohammed *et al.,* 2011b). O *i-rDNA* permite extrair com muito maior rapidez um conjunto de sequências para as quais existe uma probabilidade elevada de conterem sequências do gene *16S rRNA*, pelo que a sua utilização antes do *meta-rna* (que requer o conjunto de dados completos para análise), resulta num tempo global de processamento de cerca de 6-11 vezes inferior ao requerido para o uso do *meta-rna* apenas (Mohammed *et al.,* 2011b).

1.7.4. Alinhamento múltiplo de sequências do gene *16S rRNA*

Os alinhamentos múltiplos são uma forma de comparar e visualizar a homologia entre as sequências do gene *16S rRNA* obtidas através da sequenciação de amplicões com as sequências existentes em bases de dados. Os 3 programas descritos na **tabela 1.13** são exemplos de algoritmos usados para alinhamento múltiplo de sequências do gene *16S rRNA*. O algoritmo *Nearest Alignment Space Termination* (*NAST*) foi desenvolvido com o objetivo de desenhar sondas de sequências do gene *16S rRNA*, que pudessem ser utilizadas na tecnologia de *microarrays* para identificação de sequências deste gene em amostras desconhecidas (DeSantis *et al.*, 2006b). O *NAST* produz um alinhamento múltiplo de sequências (designado *prokMSA*) ao qual podem ser incrementadas novas sequências, facilitando os estudos de filogenia e taxonomia de procariotas (DeSantis *et al.*, 2006b). O *Python Nearest Alignment Space Termination* (*PyNAST*) (Caporaso *et al.*, 2010b) é uma reimplementação do *NAST* que permite alinhar outras sequências que não as do gene *16S rRNA* e construir alinhamentos aos pares com diferentes programas (e.g., *BLAST* ou *ClustalW*). Este algoritmo alinha uma dada sequência a um conjunto de sequências alinhadas e devolve como resultado um alinhamento múltiplo que contém o mesmo número de posições que o conjunto de alinhamentos, ou seja, em que são introduzidos *gaps* nas sequências para que a matriz de genes por posições constitua um rectângulo.

Tabela 1.13. Programas para alinhamento múltiplo de sequências do gene *16S rRNA*.

Programa	Referência	Algoritmos de alinhamento	Modelo para alinhamento	URL
NAST	DeSantis *et al.* (2006b)	Comparação de sequências comuns com 7-*mers* (alinhamento inicial), seguido de alinhamento com *BLAST*	Conjunto nuclear de 10000 sequências alinhadas da base de dados *Greengenes*	Não disponível
PyNAST	Caporaso *et al.* (2010b)	*BLAST*, *MUSCLE* (Edgar, 2004), *MAFFT* (Katoh *et al.*, 2005), *ClustalW* (Thompson *et al.*, 1994), ou através de MOM	Definido pelo utilizador	http://biocore.github.io/pynast/#
SINA	Pruesse *et al.* (2012)	Pesquisa de *k-mers* e alinhamento múltiplo de sequências usando grafos de ordem parcial (Lee *et al.*, 2002)	Base de dados *SILVA*	https://www.arb-silva.de/aligner/

1.8. Métodos de comparação de comunidades de microorganismos

Um dos principais interesses dos estudos de metagenómica é o de comparar a composição da

estrutura populacional de diferentes comunidades de microorganismos, amostradas de forma distinta no tempo e/ou no espaço. Por exemplo, podem comparar-se as comunidades de microorganismos aquáticos existentes a diferentes profundidades oceânicas ou, no caso de estudos do microbioma humano, detectar as alterações na composição populacional de um tecido após tratamento com um determinado fármaco. Para além das variações qualitativas e quantitativas dos vários táxones, é igualmente interessante perceber se as alterações da estrutura populacional reflectem alterações importantes na composição génica e/ou nas vias metabólicas dos microorganismos. Neste contexto, estão acessíveis no domínio público diversos programas para comparação de comunidades a nível estrutural e/ou funcional (designado em Inglês como "differential abundant feature detection"), dos quais estão indicados alguns exemplos na **tabela 1.14**. O *mothur* é um dos programas mais populares na área da metagenómica e, embora possa ser usado para pré-processamento e classificação de leituras do gene *16S rRNA*, também pode ser usado para comparação de amostras. Para esta finalidade, o *mothur* integra diversos programas pré-existentes, entre os quais o *SONS* (Schloss and Handelsman, 2006a), o *∫-LIBSHUFF* (Schloss *et al.*, 2004), o *TreeClimber* (Schloss and Handelsman, 2006b) e o *UniFrac* (Lozupone and Knight, 2005).

O programa *LIBSHUFF* permite determinar se as estruturas de duas comunidades ambientais de microorganismos são significativamente diferentes entre si, através da análise quantitativa de sequências do gene *16S rRNA* (Singleton *et al.*, 2001). No caso de duas amostras A e B, o método calcula a cobertura da amostra A (cobertura homóloga) como a proporção de sequências não únicas (limite de semelhança $\geq 97\%$) e a cobertura da amostra B (cobertura heteróloga) como a proporção de sequências da amostra A que estão presentes na amostra B, no total de sequências existentes na amostra A. Após o cálculo, o *LIBSHUFF* aplica a estatística de teste *Cramér-von Mises*, ou a sua forma integral no programa *∫-LIBSHUFF* (Schloss *et al.*, 2004), para determinar se existe uma diferença significativa entre os valores obtidos para as coberturas. O programa *UniFrac* (derivado da designação *Unique Fraction metric*), que também compara amostras ambientais através da análise de sequências do gene *16S rRNA*, foi desenvolvido com a premissa de que um valor fixo de semelhança, por exemplo de 98%, não é suficiente para determinar que duas quaisquer sequências que apresentem 3% ou 40% de diferenças entre si sejam tratadas da mesma forma (Lozupone and Knight, 2005). O método *UniFrac* permite a comparação de 3 ou mais amostras em simultâneo, assim como considera o comprimento do ramo da árvore filogenética nessa mesma comparação. O *TreeClimber* é um outro método filogenético que utiliza um teste baseado na parsimonia para avaliar se duas ou mais comunidades de microorganismos partilham a mesma estrutura composicional (Schloss and Handelsman, 2006b).

Os métodos de comparação de comunidades acima descritos podem considerar-se essencialmente como métodos estatísticos comuns aplicados a dados de sequências do gene *16S rRNA*, mas que não têm a preocupação de avaliar de forma fina as comunidades do ponto de vista biológico. Por outras palavras, o que se pretende determinar quando se obtém um resultado estatisticamente significativo, é se esse resultado é consequência de artefactos de amostragem ou de diferenças reais na taxonomia ou ecologia das comunidades. O programa *SONS* foi um dos primeiros a penetrar neste nível de análise, ao avaliar a riqueza e a fracção de UTO que é partilhada por duas comunidades de microorganismos, através do cálculo de estimadores não-paramétricos (Schloss and Handelsman, 2006a). O método designado por *Linear Discriminant Analysis* (LDA) *effect size* (*LefSe*) procura determinar as características dos dados de metagenómica (por exemplo, microorganismos ou genes) que são responsáveis pelas diferenças entre 2 ou mais classes (Segata *et al.*, 2011). Primeiramente são aplicados testes estatísticos que avaliam a existência de diferenças significativas em termos de características entre classes. Estas características são também avaliadas estatisticamente em termos de

uma hipótese "biológica" e as diferenças significativas são avaliadas em termos da sua magnitude. Numa segunda fase, os biomarcadores identificados podem ser visualizados em árvores taxonómicas ou funcionais, o que facilita a interpretação dos resultados (Segata *et al.*, 2011).

Tabela 1.14. Programas para comparação de comunidades.

Programa	Referência	Descrição	URL
LefSe	Segata *et al.* (2011)	Junta testes convencionais para determinação da significância estatística com testes que avaliam a consistência biológica e a relevância do efeito	http://huttenhower.sph.harvard.edu/galaxy/
MetaBoot	Wang *et al.* (2015)	Baseado nos perfis taxonómicos gerados pela análise do gene *16S rRNA*, a partir dos quais são seleccionados um conjunto de genes informativos utilizando o método de "minimum redundancy - maximum relevance" aplicado a dados de *microarrays* (Ding and Peng, 2005)	http://www.computationalbioenergy.org./metaboot.html
MetaStats	White *et al.* (2009)	Utiliza contagens de leituras de sequenciação *shotgun* ou do gene *16S rRNA* de múltiplas amostras, relativas a diferentes condições experimentais, para diferentes características (por ex., vias metabólicas), e avalia estatisticamente as diferenças usando uma nova medida de confiança	http://metastats.cbcb.umd.edu/
mothur	Schloss *et al.* (2009)	Incorpora os programas *SONS*, ∫*-LIBSHUFF*, *TreeClimber* e *UniFrac* para comparação de amostras	https://www.*mothur*.org
STAMP	Parks *et al.* (2014)	Importa perfis taxonómicos e funcionais gerados por outros programas (e.g., *QIIME*), aplica testes estatísticos de hipóteses, determina o tamanho do efeito e os intervalos de confiança, e produz gráficos para as características seleccionadas	http://kiwi.cs.dal.ca/Software/STAMP

1.9. Métodos de anotação funcional de comunidades de microorganismos

Um dos passos seguintes à análise taxonómica de uma comunidade de microorganismos é, frequentemente, a caracterização do seu perfil funcional. Em função dos microorganismos que a constituem, e do nicho ambiental que ocupa, cada comunidade conterá um *pool* de genes que lhe permite adaptar-se com sucesso ao ambiente circundante. Este *pool* de genes poderá estar associado a determinadas funções bioquímicas como, por exemplo, a degradação de glicosaminoglicanos no intestino humano (Abubucker *et al.*, 2012). Uma forma de conhecer o perfil funcional de comunidades de microorganismos é inferi-lo a partir da sequenciação de metagenomas ou de amplicões do gene *16S rRNA*, usando algoritmos bioinformáticos apropriados a cada tipo de dados. Na **tabela 1.15** apresentam-se alguns programas que podem ser usados para anotação funcional, embora alguns dos programas referidos nas secções anteriores (por exemplo, o *MEGAN4*) também permitam obter perfis funcionais. Inversamente, alguns dos programas referidos na tabela também permitem realizar a análise taxonómica em paralelo com a análise funcional, como é o caso do *Parallel-META* (Su *et al.*, 2014a).

1.9.1. Anotação funcional com base em leituras de sequenciação *shotgun*

A sequenciação *shotgun* de metagenomas oferece o conjunto de dados preferencial para inferir o perfil funcional de uma comunidade de microorganismos, uma vez que permite obter, pelo menos para

as espécies mais abundantes, a sequência da maioria dos genes que são parte dos respectivos genomas. Neste âmbito, existem 4 programas que podem ser usados e que incluem o *Functional and taxonomic analysis of metagenomes* (*FANTOM*) (Sanli *et al.*, 2013), o *Functional Ontology Assignments for Metagenomes* (*FOAM*) (Prestat *et al.*, 2014), o *Parallel-META* e o *COGNIZER* (Bose *et al.*, 2015). O *software FANTOM* permite obter informação funcional hierárquica a partir da integração dos dados de um ficheiro de abundâncias (contendo anotações taxonómicas ou funcionais) e de um ficheiro de metadados (contendo propriedades das amostras), com bases de dados de funções génicas como o *KEGG* (Kanehisa and Goto, 2000), de classificação filogenética de proteínas como o *Clusters of Orthologous Groups* (*COG*; Tatusov *et al.*, 2000), e as bases de dados de famílias proteicas *Pfam* (Finn *et al.*, 2010) e *TIGRFAMs* (Haft *et al.*, 2003).

Tabela 1.15. Programas para anotação funcional de comunidades.

Programa	Referência	Tipo de dados	Descrição	URL
COGNIZER	Bose *et al.* (2015)	*shotgun*	Anotação funcional de dados de sequenciação metagenómica e metatranscritómica usando cruzamento de bases de dados (*COG*, *KEGG*, *SEED*, *GO* e *Pfam*)	http://metagenomics.atc.tcs.com/cognizer/
FANTOM	Sanli *et al.* (2013)	*shotgun*, *16S rRNA*	Integração de dados de sequenciação com dados taxonómicos (*NCBI Taxonomy*) e dados funcionais (*KEGG*, *COG*, *Pfam* e *TIGRFAMs*)	http://www.sysbio.se/Fantom/
FOAM	Prestat *et al.* (2014)	*shotgun*	Identificação de genes pertencentes à mesma família funcional usando MOM e descrição dos grupos funcionais usando uma ontologia própria de 5 níveis hierárquicos	http://cbb.pnnl.gov/portal/software/FOAM.html
Parallel-META	Su *et al.* (2014a)	*shotgun*, *16S rRNA*, *18S rRNA*	Classificação taxonómica (nos genes *16S rRNA* e *18S rRNA*) e análise funcional (*GO* e *SEED*) de dados de sequenciação	http://www.computationalbioenergy.org/parallel-meta.html
PICRUSt	Langille *et al.* (2013)	*shotgun*, *16S rRNA*	Classificação funcional com base em *KO´s* e *COG´s* usando algoritmos de reconstrução de estados ancestrais	http://picrust.github.io/picrust/
Tax4Fun	Asshauer *et al.* (2015)	*shotgun*, *16S rRNA*	Anotação funcional com base no *KEGG* usando o menor ancestral comum e a semelhança das sequências	http://tax4fun.gobics.de/
Vikodak	Nagpal *et al.* (2016)	*16S rRNA*	Anotação de vias metabólicas e enzimas com base em dados do *PATRIC*, *IMG* e *KEGG*	http://metagenomics.atc.tcs.com/vikodak/

O *FOAM* constitui uma base de dados de informação génica funcional, que foi desenvolvida com o intuito de analisar dados de sequenciação de metagenómica e metatranscriptómica ambientais (Prestat *et al.*, 2014). A informação funcional é alimentada pela base de dados *KEGG orthologs* (*KO*; Kanehisa *et al.*, 2008), a qual fornece a lista de genes homólogos contidos no *KEGG* que correspondem à mesma família funcional (Prestat *et al.*, 2014). O *FOAM* permite identificar os *KO* através de MOM. Adicionalmente, o *FOAM* contém uma ontologia funcional, hierarquizada em 5 níveis, que foi criada para descrever os grupos funcionais e que alberga as principais funções bioquímicas e vias metabólicas associadas com a microbiologia ambiental.

No *Parallel-META* as leituras da sequenciação *shotgun* são primeiramente montadas usando o *Velvet*, seguido de previsão de genes utilizando o programa *FragGeneScan* (ver secção 1.10), que incorpora modelos de erros de sequenciação e selecção de codões num MOM para identificar genes codificantes de proteínas em leituras de tamanho curto (Rho *et al.*, 2010). A identificação das

sequências génicas assim obtidas é efectuada através do alinhamento nas bases de dados de referência *nr* e de sub-sistemas da *SEED* (Overbeek *et al.*, 2005), sendo as respetivas anotações funcionais recolhidas a partir da estrutura hierárquica dos termos do *Gene Ontology* (GO) (Holliday *et al.*, 2017) e dos 4 níveis que compõem a SEED (Su *et al.*, 2014a). De acordo com Overbeek *et al.* (2005), um sub-sistema é definido como "um conjunto de papeís funcionais que no seu todo implementam um processo ou um complexo estrutural biológico específico" e de que é exemplo o ribossoma.

O *COGNIZER* é um programa que pode ser usado em conjuntos de dados de sequenciação metagenómica ou metatranscriptómica, e que permite obter anotações funcionais de diversas fontes de informação incluindo o *COG*, *KEGG*, *SEED*, *GO* e *Pfam* (Bose *et al.*, 2015). O *COGNIZER* dispõe de uma base de dados própria que contém as relações cruzadas entre o *COG* e as outras bases de dados funcionais. O algoritmo permite efectuar uma pesquisa de homologia contra as sequências da base de dados *COG* usando o *BLASTX* ou o *RAPSearch* (Ye *et al.*, 2011), sendo cada leitura atribuída a uma das 25 categorias funcionais principais do *COG* de acordo com o valor mais elevado de homologia. Seguidamente, cada leitura é anotada funcionalmente usando a base de dados do *COGNIZER* (Bose *et al.*, 2015). As anotações obtidas são então cruzadas com as restantes anotações para inferir a função.

1.9.2. Anotação funcional com base em leituras do gene *16S rRNA*

Apesar da sequenciação de genes marcadores, como o gene *16S rRNA*, não permitir obter informação sobre os restantes genes presentes num genoma, é possível inferir, de forma indirecta, o perfil funcional do respectivo metagenoma. No domínio público existem pelo menos 3 programas que podem ser utilizados para anotação funcional com base em leituras do gene *16S rRNA* e que incluem o *phylogenetic investigation of communities by reconstruction of unobserved states* (*PICRUSt*) (Langille *et al.*, 2013), o *Tax4Fun* (Asshauer *et al.*, 2015) e o *Vikodak* (Nagpal *et al.*, 2016). O programa *PICRUSt* é um programa que explora a associação entre a filogenia e a função para prever a composição funcional (sobretudo com base em *KO* e *COG)* de um metagenoma a partir de dados do gene *16S rRNA*. O princípio desta associação é a possibilidade de prever as funções codificadas pelo genoma de um determinado organismo, conhecendo as funções que são codificadas por organismos relacionados evolutivamente (Langille *et al.*, 2013). O primeiro passo deste método (realizado uma única vez) envolve a previsão da abundância de famílias de genes para cada organismo, cuja sequência genómica está disponível numa base de dados de referência, de acordo com uma árvore filogenética construída com base em informação do gene *16S rRNA*. O conteúdo génico de cada microorganismo no metagenoma é previsto inferindo o conteúdo do último ancestral filogenético comum (para o qual exista pelo menos 1 genoma sequenciado), através de algoritmos de reconstrução de estados ancestrais. De seguida, a proporção das famílias génicas na comunidade é determinada combinando os dados de abundância génica com as proporções relativas de cada táxon determinadas pela sequenciação do gene *16S rRNA*, as quais são normalizadas em função do número de cópias deste gene em cada genoma de referência (Langille *et al.*, 2013).

À semelhança do *PICRUSt*, o programa *Tax4Fun* também baseia a classificação do perfil funcional com base na pesquisa do ancestral mais próximo (Asshauer *et al.*, 2015). O método é iniciado com o agrupamento das leituras do gene *16S rRNA* e a sua atribuição a sequências de referência existentes na base de dados *SILVA*. A composição da comunidade é transformada num perfil taxonómico de procariotas do *KEGG*, que é depois normalizado de acordo com o número de cópias do gene *16S rRNA* (Asshauer *et al.*, 2015). Finalmente, o perfil funcional é obtido com base nas abundâncias relativas dos vários microorganismos. O *Tax4Fun* tem em conta um mínimo de semelhança da

sequência do gene *16S rRNA* para ligar todos os táxones na árvore filogenética, enquanto o *PICRUSt* não considera a distância genética na ligação entre os táxones (Asshauer *et al.*, 2015).

O programa *Vikodak* é uma alternativa aos programas acima descritos, estando assente numa base de dados de enzimas (ligadas às vias metabólicas do *KEGG*) e respectivo número de cópias génicas presente no genoma de cada microorganismo. Esta base de dados foi construída com base na informação disponível na *IMG* e na base de dados *PATRIC* (Wattam *et al.*, 2014), que contém dados de genomas e estruturas proteícas tri-dimensionais de bactérias. O *Vikodak* está organizado em 3 módulos de análise distintos que permitem estimar o predomínio das várias vias metabólicas numa determinada amostra (e a contribuição de cada microorganismo para as funções dessas vias), comparar estatisticamente os perfis de várias comunidades e detalhar o perfil enzimático das vias metabólicas em cada comunidade ou entre comunidades (Nagpal *et al.*, 2016).

1.10. Métodos de previsão de genes em metagenomas

Uma das finalidades associadas ao estudo de metagenomas é a identificação de sequências génicas a partir de fragmentos genómicos de tamanho reduzido, a qual pode ser conseguida usando um dos programas listados na **tabela 1.16**. A identificação de genes pode ser realizada através da pesquisa de homologia com sequências depositadas em bases de dados, mas não permite o reconhecimento de novos genes, os quais podem estar associados a um perfil funcional específico. Os MOM têm sido utilizados para a identificação de genes em sequências genómicas, mas requerem à partida o conhecimento de um grande número de sequências génicas para aprendizagem dos parâmetros de espécies individuais (revisto em Nogushi *et al.*, 2006). Dado que existe uma correlação entre o conteúdo GC das sequências genómicas de procariotas e a frequência dos vários codões (Karlin, 2001), é possível criar modelos que permitem determinar os codões utilizados num dado genoma (Nogushi *et al.*, 2006) e, consequentemente, possibilitar a identificação de genes. Por exemplo, o programa *Metagene* utiliza a frequência de di-codões em vez dos codões individuais para obter uma maior precisão na previsão de genes. Adicionalmente, o *Metagene* utiliza a distribuição de frequências dos comprimentos das *open reading frames* (ORFs), a distância aos codões de iniciação e as distâncias entre ORFs vizinhas como medidas para aumentar a especificidade das previsões (Nogushi *et al.*, 2006).

Tabela 1.16. Programas para previsão de genes em metagenomas.

Programa	Referência	Descrição	URL
FragGeneScan	Rho *et al.* (2010)	Modelo probabilístico baseado em MOM que combina erros de sequenciação com a utilização de codões	https://sourceforge.net/projects/fraggenescan/
Glimmer-MG	Kelley *et al.* (2012)	Baseado em agrupamentos de sequências e modelos de erros de sequenciação (substituições e *indels*)	http://www.cbcb.umd.edu/software/glimmer-mg/
MetaGene	Noguchi *et al.* (2006)	Combina a frequência de di-codões, distribuição dos comprimentos das ORFs, distância aos codões de iniciação e as distências entre ORFs adjacentes	http://metagene.cb.k.u-tokyo.ac.jp/
RAMMCAP	Li (2009)	Determina as sequências codificantes (programa *ORF_finder*) após agrupamento de leituras	http://weizhong-lab.ucsd.edu/rammcap/cgi-bin/rammcap.cgi

As leituras de comprimento reduzido, como as produzidas pelas plataformas de sequenciação *Illumina* e *Ion Torrent*, bem como os erros de sequenciação (*indels*) podem causar alteração da grelha de leitura, dificultando assim a localização de sequências codificantes. Neste sentido, foi desenvolvido

o programa *FragGeneScan* que utiliza um modelo probabilístico baseado num MOM, o qual combina erros de sequenciação com utilização de codões para fazer a previsão de genes em leituras de comprimento tão curto quanto 100 pb (Rho *et al.*, 2010). Uma vez que o *FragGeneScan* permite prever genes cuja sequência não começa num codão de iniciação, ou que não termina num codão de terminação, pode ser utilizado em leituras que correspondam a sequências génicas incompletas.

1.11. Selecção de programas para identificação taxonómica de leituras com base no número de citações

Nas secções anteriores descreveram-se as diferentes etapas de análise de leituras de sequenciação *shotgun* ou do gene *16S rRNA*, aplicadas a estudos de metagenómica, assim como uma parte significativa dos programas disponibilizados no domínio público entre 2005 e 2016, que podem ser usados para essa finalidade. De entre um total de 142 programas mencionados no texto deste trabalho, são particularmente importantes os programas que permitem identificar os diferentes microorganismos que compõem um metagenoma. Nas secções 1.4, 1.5, 1.6 e 1.7.2 foram descritos 91 programas que podem ser usados para a caracterização taxonómica de uma comunidade de microorganismos, a partir de leituras de sequenciação *shotgun* ou de amplicões do gene *16S rRNA*. Por forma a avaliar a relevância destes programas nos estudos de metagenómica publicados na literatura científica, fez-se uma análise do número de citações dos artigos da descrição original de cada um dos programas, recorrendo-se à informação disponível na base de dados *Web of Science* (Web of Science, 2014). Nesta análise foram igualmente incluídos os programas *mothur* e *Parallel-META*, uma vez que estes também podem ser utilizados para identificação taxonómica de leituras. O número total de citações por artigo até final de 2016 encontra-se representado graficamente na **figura 1.5**. É possível constatar desta análise que apenas 5 dos 91 programas (5%), incluindo o *mothur*, *QIIME*, *RDP Classifier*, *DOTUR* e *SINA*, têm um número total de citações superior a 500, enquanto cerca de 73% dos programas têm menos de 50 citações cada. Em particular, os 3 programas mais citados (*mothur*, *QIIME* e *RDP Classifier*) têm, globalmente, mais do dobro das citações (12907) do que todos os restantes 88 programas em conjunto (5928).

É perceptível na análise do gráfico exterior que os programas *mothur* e *QIIME* têm um número de citações que se destaca claramente dos restantes programas. Uma vez que estes programas datam de 2009 (*mothur*) e 2010 (*QIIME*), analisou-se a variação do número anual de citações nos últimos 4 anos completos (2013 a 2016). De acordo com o gráfico interior, verifica-se que estes 2 programas têm vindo a ter uma utilização crescente nos últimos anos e que o *QIIME* suplantou de forma significativa o *mothur* no número de citações anual a partir de 2015 (inclusive), sendo actualmente o programa mais usado em estudos de metagenómica. Em síntese, os programas *mothur* e *QIIME* são os programas de identificação taxonómica de leituras mais citados na literatura, pelo que podem ser considerados como ferramentas bioinformáticas de referência em metagenómica, para análise de dados de sequenciação do gene *16S rRNA*. Neste contexto, foi implementado no presente trabalho um programa em linguagem *Matlab*, designado *sim16S*, que permite criar conjuntos de leituras simuladas do gene *16S rRNA*, as quais podem ser usadas para avaliação *benchmarking* de programas de classificação taxonómica de leituras. Neste trabalho, a performance de classificação dos programas *QIIME* e *mothur* foi testada usando diversos conjuntos de leituras simuladas produzidas pelo *sim16S*.

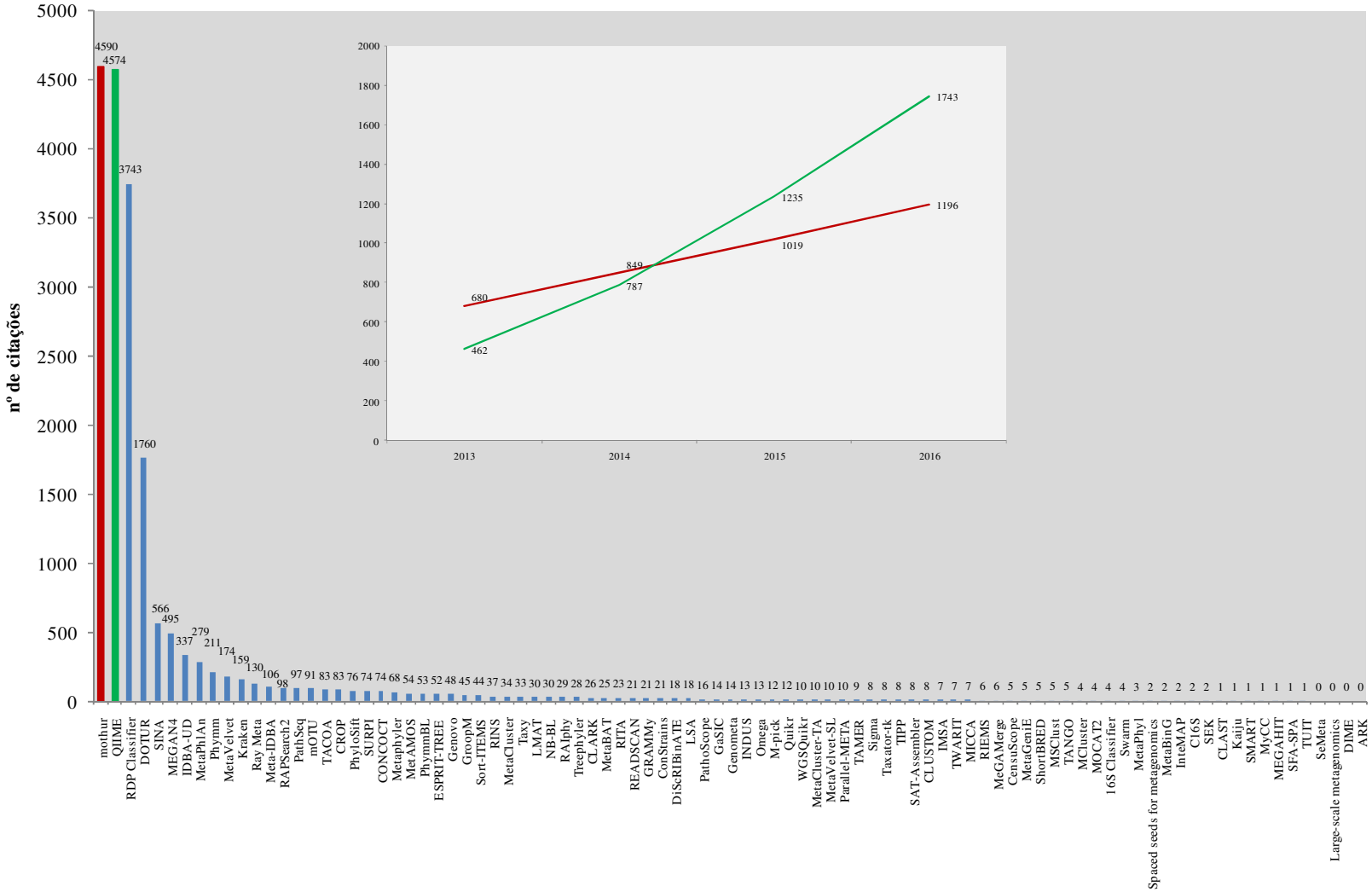

Figura 1.5. Representação gráfica do número total de citações de 91 artigos referentes a métodos de identificação taxonómica de leituras de sequenciação *shotgun* ou de amplicões do gene *16S rRNA*. O gráfico interior representa a variação do número de citações ao longo dos anos 2013 a 2016 para os programas *mothur* (linha vermelha) e *QIIME* (linha verde). Os dados de citações foram obtidos a partir da informação disponível na base de dados *Web of Science* e incluem os anos 2005 a 2016 inclusive.

2. Objectivos

O objectivo principal deste trabalho é o de avaliar a performance de ferramentas computacionais de referência, usadas para análise de dados de sequenciação em metagenómica. Neste âmbito, são propostos os 2 seguintes objectivos específicos:

1) Criação e implementação de um programa informático para simulação de leituras de sequenciação do gene *16S rRNA*;

2) Comparação dos resultados da classificação taxonómica dos programas *QIIME* e *mothur*, obtidos com base em conjuntos de leituras simuladas do gene *16S rRNA*.

3. Métodos

3.1. Simulação de leituras do gene *16S rRNA* (programa *sim16S*)

O *software Matlab* (versão R2016a) foi usado para criar um programa (designado *sim16S*, abreviatura obtida de **sim**ulador de leituras do gene ***16S** rRNA*), que simula leituras do gene *16S rRNA* idênticas às obtidas numa plataforma de sequenciação de nova geração. Além da produção das leituras, o programa permite também introduzir substituições de bases de forma a simular a ocorrência de erros de sequenciação. Desta forma, é possível testar diferentes *softwares* de classificação taxonómica de leituras do gene *16S rRNA* e verificar se e como a classificação poderá ser afectada por aqueles erros. O programa *sim16S* é constituído por 8 funções sequenciais, que vão desde a recolha de sequências do gene *16S rRNA* a partir de uma base de dados até à emissão de um relatório (**figura 3.1**). O programa foi estruturado nas 3 etapas abaixo descritas que pretendem simular, sucessivamente, uma reacção de amplificação de sequências do gene *16S rRNA*, a composição de uma amostra de uma comunidade de microorganismos e o processo de sequenciação das leituras dessa amostra.

Etapa 1: Selecção de táxones com base em sequências de amplicões do gene *16S rRNA* (funções *seqConc.m*, *seqAmplicon.m* e *compareAmp.m*).

A primeira função do programa (*seqConc.m*) consiste na concatenação das linhas da sequência dos genes *16S rRNA* existentes no ficheiro da base de dados, produzindo ficheiros com as sequências de referência (*refSeq.fasta*) e respectiva taxonomia (*taxonomyRef.txt*). O número total de sequências é especificado pelo utilizador. Esta função pode ser aplicada uma única vez para cada base de dados de sequências de referência, diminuindo assim o tempo total de computação do programa. Seguidamente, o programa utiliza a função *seqAmplicon.m* para extrair amplicões do gene *16S rRNA* a partir das sequências de 2 oligonucleótidos iniciadores específicos (*forward* e *reverse*) introduzidas pelo utilizador. Esta função produz ficheiros com as sequências dos amplicões seleccionados (*ampliconSet.txt*) e respectivas taxonomias (*speciesSet.txt*), assim como um ficheiro com as sequências que não têm homologia com um ou ambos os oligonucleótidos (*excludedSeq.fasta*). Uma vez que alguns dos amplicões poderão apresentar uma sequência total idêntica entre si, a função *compareAmp.m* utiliza o ficheiro *ampliconSet.txt* para calcular o número de vezes que cada amplicão se encontra repetido no conjunto total de amplicões (ficheiro *repeatSeq.txt*). Esta informação pode ser útil para perceber se a classificação taxonómica de um dado amplicão pode ser diferente do esperado, apenas porque a mesma sequência pode estar presente em grupos taxonómicos diferentes.

Etapa 2: Produção de um conjunto de 10000 leituras seleccionadas aleatoriamente (funções *randAmp.m* e *countTaxa.m*).

A função *randAmp.m* produz 10000 valores inteiros a partir de uma distribuição de *Poisson*, em que o parâmetro *lambda* é igual a metade do número de amplicões disponíveis, e coloca estes valores numa matriz. A contagem do número de vezes que um determinado valor inteiro está presente na matriz irá definir a proporção do respectivo amplicão num conjunto total de 10000 leituras. A ideia deste método é a de simular a estrutura típica de uma comunidade de microorganismos, em que o espectro de abundâncias é muito variável, compreendendo alguns táxones muito abundantes e outros residuais (**figura 3.2**). A função dá origem a 2 ficheiros que vão incluir o número de amplicões de cada

Programa *sim16S*

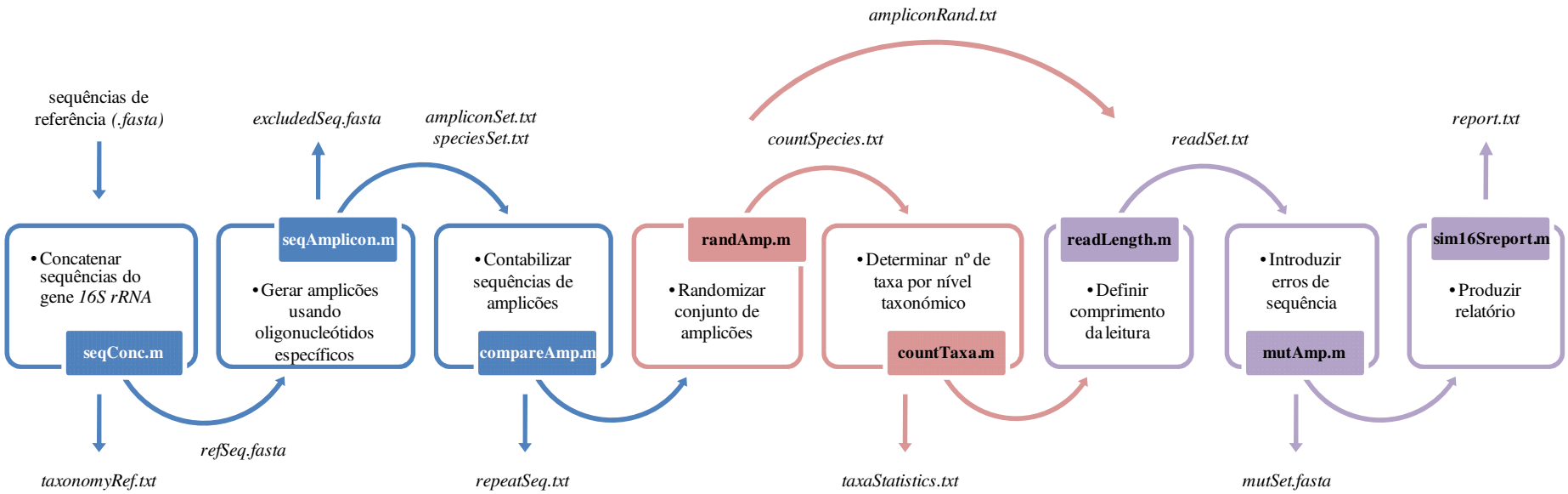

Figura 3.1. Esquema representativo das várias etapas e funções do programa *sim16S*. O programa contém 8 funções sequenciais, indicadas nas caixas coloridas, que se distribuem por 3 etapas. O programa inicia-se com a concatenação das sequências de referência da base de dados *SILVA* (ficheiro *fasta*) e termina com a produção do relatório (ficheiro *report.txt*). Os ficheiros produzidos por cada uma das funções estão indicados junto às respectivas setas.

táxon (*countSpecies.txt*) após randomização e o conjunto total de 10000 sequências de amplicões (*ampliconRand.txt*). Seguidamente, a função *countTaxa.m* permite calcular, para os níveis taxonómicos de filo, classe, ordem, família, género e espécie, o número de vezes que cada grupo taxonómico está presente no conjunto de amplicões (ficheiro *taxaStatistics.txt*). Esta informação é útil para comparar de forma fácil os resultados obtidos com os diferentes *softwares* de classificação taxonómica, uma vez que estes permitem agrupar o número de sequências classificadas por grupo taxonómico.

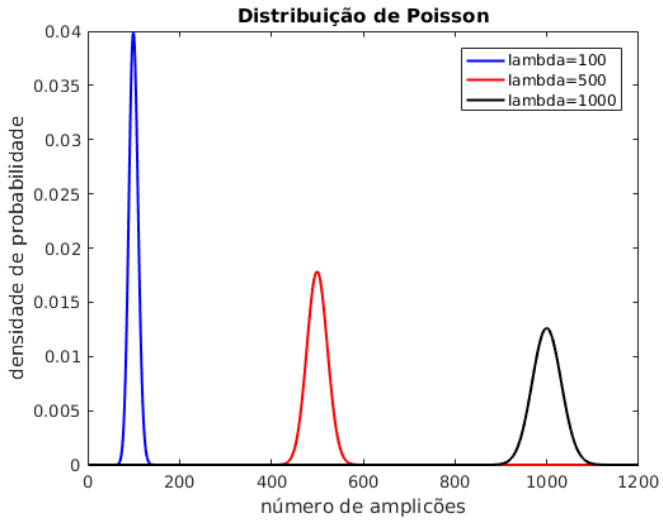

Figura 3.2. Gráfico da função de densidade de probabilidade para a distribuição de *Poisson*, usando diferentes valores do parâmetro *lambda*. O aumento do valor de *lambda* (i.e., número de amplicões) conduz a um aumento da base da curva da função. Desta forma, a variação do valor de *lambda* permite obter diferentes contagens para cada amplicão após selecção aleatória de 10000 valores com base na distribuição de *Poisson*, o que permite simular diferentes estruturas composicionais de uma comunidade de microorganismos. O gráfico foi elaborado usando o *script Matlab* descrito no **Anexo A**.

Etapa 3: Introdução de substituições de bases nas leituras do conjunto de dados de acordo com uma função estatística que simula os erros de sequenciação (funções *readLength.m*, *mutAmp.m* e *sim16Sreport.m*).

A função *readLength.m* usa o ficheiro *ampliconRand.txt* para limitar o comprimento das sequências a um valor fixo (ficheiro *readSet.txt*), de forma a simular o comprimento das leituras obtidas nas plataformas de sequenciação. Seguidamente, o programa usa a função *mutAmp.m* que permite introduzir substituições de bases nas leituras de acordo com especificações introduzidas pelo utilizador. Em particular, o utilizador pode definir o número de substituições que ocorrem em cada leitura e a proporção de leituras que contêm essas substituições. Desta forma, é possível gerar diferentes conjuntos de dados simulados com uma proporção variável de erros de sequenciação, o que é útil para testar a performance de diferentes programas e algoritmos de análise de leituras do gene *16S rRNA*. A posição da substituição é escolhida aleatoriamente em cada leitura individual de acordo com uma distribuição *Half-Normal* adaptada ao comprimento da leitura. Os valores do parâmetro *sigma* foram escolhidos para que a probabilidade se aproximasse de zero na última posição de cada leitura (150 ou 250 bases). No entanto, uma vez que existe uma probabilidade muito reduzida de ser escolhida uma posição de substituição maior que o comprimento da leitura, optou-se por truncar a distribuição *Half-Normal* entre a primeira e última posições das leituras (inclusive), para que a probabilidade cumulativa seja de 1 entre os 2 valores (**figura 3.3**). Uma vez que de acordo com esta distribuição, a probabilidade de ocorrência de uma substituição é maior no início da leitura do que no

final, a função *mutAmp.m* corrige a posição da substituição usando a última posição da leitura como o início da mesma (posição 1). Esta distribuição de erros pretende reproduzir a perda gradual de qualidade das leituras que se observa, por exemplo, nas plataformas *Illumina*, ao longo do comprimento da leitura. A função *mutAmp.m* produz o ficheiro principal (*mutSet.fasta*) do programa *sim16S*, que contém 10000 sequências nas quais foram introduzidas as substituições de bases. As sequências estão identificadas por um número de ordem (seq1, seq2, seq3, etc.). Este ficheiro é utilizado para classificação taxonómica de leituras do gene *16S rRNA* (ver secção 3.2). Por fim, a função *sim16Sreport.m* produz um relatório (ficheiro *report.txt*) com os dados da execução do programa, incluindo os parâmetros introduzidos pelo utilizador e os dados gerados pelo programa com base naqueles parâmetros. As funções *Matlab* do programa *sim16S* estão descritas no **Anexo C**. A **tabela 3.1** descreve cada um dos ficheiros gerados pelo programa *sim16S*.

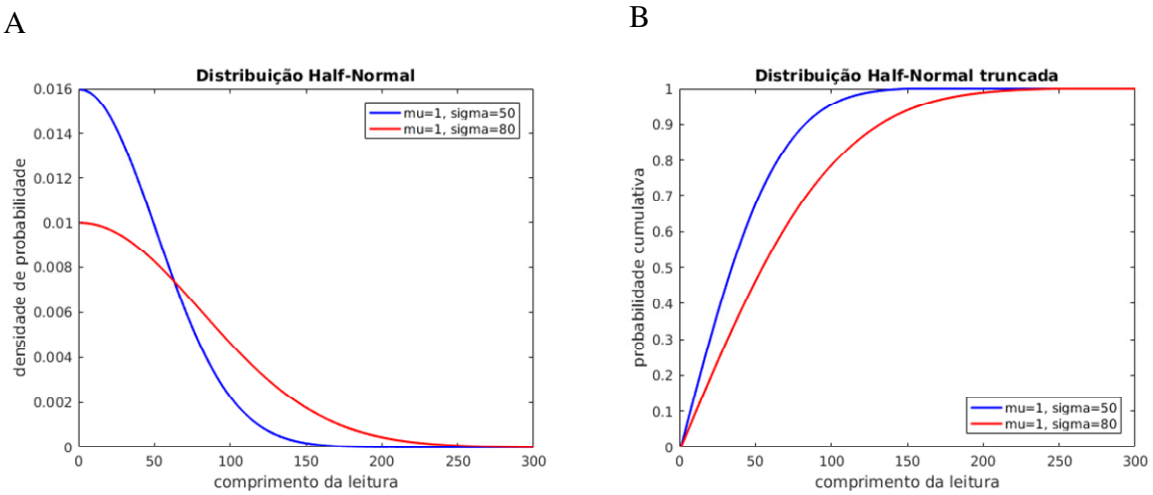

Figura 3.3. Gráficos da função de densidade de probabilidade para a distribuição *Half-Normal* (A) e da função de probabilidade cumulativa para a distribuição *Half-Normal* truncada (B), usando diferentes valores do parâmetro *sigma*. De acordo com a distribuição *Half-Normal*, a probabilidade do acontecimento (ou seja, a substituição de uma base da sequência) decresce gradualmente ao longo do comprimento da leitura para os diferentes valores de *sigma*. Uma vez que as leituras obtidas no programa *sim16S* têm comprimento pré-definido (150 pb ou 250 pb), a distribuição foi truncada nos limites de cada comprimento de leitura para que a probabilidade cumulativa fosse de 1 dentro do comprimento de cada leitura. Os gráficos foram elaborados usando o *script Matlab* descrito no **Anexo B**.

O programa *sim16S* é executado na janela de comandos do *Matlab* usando a seguinte expressão:
>> *sim16S*(*fastaFile*, *numSeq*, *primerF*, *primerR*, *readLen*, *mutReads*, *numMut*)

Os 7 argumentos da função *sim16S* (e respetivos valores) são os seguintes:

1) *fastaFile*: Base de dados de sequências do gene *16S rRNA* (valor: Nome do ficheiro da base de dados em formato *fasta*);

2) *numSeq*: Total de sequências a extrair da base de dados (valor: Número inteiro entre 1 e 597607);

3) *primerF*: Sequência do oligonucleótido *forward* usado para selecção de um amplicão específico do gene *16S rRNA*. (valor: Sequência de bases introduzida no sentido 5´-3´);

4) *primerR*: Sequência do oligonucleótido *reverse* usado para selecção de um amplicão específico do gene *16S rRNA*. (valor: Sequência de bases introduzida no sentido 5´-3´);

5) *readLen*: Comprimento máximo das leituras de sequenciação (valor: 150 ou 250);

6) *mutReads*: comprimento do intervalo a partir do qual é determinada a proporção de leituras com erros de sequenciação (valor: Número inteiro positivo). Quanto maior o comprimento do intervalo, menor a probabilidade de que uma leitura contenha erros de sequenciação. Por exemplo, se *mutReads*=100, então o intervalo conterá 100 valores inteiros positivos e a probabilidade de se originar 1 leitura mutada no conjunto total de leituras é de 1%.

7) *numMut*: Número de substituições a introduzir em cada leitura (valor: Número inteiro positivo).

Tabela 3.1. Descrição dos ficheiros gerados pelo programa *sim16S*.

Nome do ficheiro	Descrição
refSeq.fasta	Sequências em formato *fasta* geradas a partir da base de dados de sequências de referência
taxonomyRef.txt	Taxonomia das sequências contidas no ficheiro *refSeq.fasta* incluindo número de acesso e descrição taxonómica
ampliconSet.txt	Sequências dos amplicões seleccionados com base nos oligonucleótidos *forward* e *reverse*
speciesSet.txt	Taxonomia das sequências correspondentes aos amplicões do ficheiro *ampliconSet.txt*
excludedSeq.fasta	Sequências em formato *fasta* nas quais não foram seleccionados amplicões
repeatSeq.txt	Número de ocorrências de sequências idênticas de amplicões por cada sequência seleccionada
countSpecies.txt	Número total de cada táxon no conjunto de sequências randomizadas (total=10000)
ampliconRand.txt	Sequências dos amplicões após randomização (total=10000)
taxaStatistics.txt	Contagem do número total de cada grupo taxonómico (filo a espécie) presente no ficheiro *ampliconRand.txt*
readSet.txt	Sequências dos amplicões com comprimento da leitura pré-definido (150 pb ou 250 pb)
mutSet.fasta	Sequências dos amplicões com substituição de 1 ou mais bases numa dada proporção de leituras
report.txt	Relatório da execução do programa *sim16S* incluindo dados introduzidos pelo utilizador e gerados pelo programa

Os amplicões do gene *16S rRNA* usados para a simulação de leituras foram obtidos usando as sequências de oligonucleótidos descritas na **tabela 3.2**. Os pares de oligonucleótidos escolhidos cobrem as regiões hipervariáveis 3 e 4 e são apropriados para as plataformas de sequenciação *Illumina* e *Ion Torrent*, uma vez que geram amplicões de tamanho reduzido (< 300 pb). De acordo com Klindworth *et al.* (2013), o par de oligonucleótidos S-D-Bact-0341-b-S-17/S-D-Bact-0515-a-A-19 é específico do domínio *Bacteria* e cobre 91.2% das sequências da base de dados *SILVA* (versão *SSURef 108 NR)*, enquanto o par S-D-Arch-0519-a-S-15/S-D-Bact-0785-b-A-18 cobre 88% das sequências do domínio *Archaea* e 89.1% do domínio *Bacteria*.

Tabela 3.2. Descrição dos oligonucleótidos usados para simulação de amplicões do gene *16S rRNA*.

Amplicão	Designação dos oligonucleótidos (a)	Domínio	Região hipervariável	Direcção	Sequência (5´-3´) (b)	Tamanho aproximado (pb) (c)
A	S-D-Bact-0341-b-S-17	*Bacteria*	V3	*forward*	CCTACGGGNGGCWGCAG	193
	S-D-Bact-0515-a-A-19			*reverse*	TTACCGCGGCTGCTGGCAC	
B	S-D-Arch-0519-a-S-15	*Archaea* e *Bacteria*	V4	*forward*	CAGCMGCCGCGGTAA	284
	S-D-Bact-0785-b-A-18			*reverse*	TACNVGGGTATCTAATCC	

(a) A designação dos oligonucleótidos respeita a descrição de Klindworth *et al.* (2013); (b) Dadas as posições de base mista existentes nos oligonucleótidos S-D-Bact-0341-b-S-17 (bases N e W), S-D-Arch-0519-a-S-15 (base M) e S-D-Bact-0785-b-A-18 (bases N e V), optou-se por utilizar as sequências 5'-CCTACGGGAGGCAGCAG-3', 5'-CAGCAGCCGCGGTAA-3' e 5'-TACCAGGGTATCTAATCC-3', respectivamente, uma vez que estas estavam mais representadas da base de dados de sequências de referência utilizada; (c) O tamanho dos amplicões foi obtido de Klindworth *et al.* (2013).

A base de dados *SILVA* (ficheiro *SILVA_123_SSURef_Nr99_tax_silva.fasta*) foi usada para obter as sequências do gene *16S rRNA* e a respectiva taxonomia (Quast *et al.*, 2013, Yilmaz *et al.*, 2014). A versão 123 da *SILVA* é composta por 597607 sequências não redundantes dos genes da subunidade pequena do RNA ribossomal. Esta base de dados foi construída a partir da base de dados completa de sequências de referência usando o programa *UCLUST* (Edgar, 2010) para remover sequências redundantes que apresentam ≥ 99% de identidade entre si, ou seja, trata-se de uma base de dados de

sequências representativas dos genes da subunidade pequena do RNA ribossomal. No programa *sim16S*, o número máximo de sequências analisadas foi apenas de 5000, de forma a reduzir o tempo de processamento computacional. Os conjuntos de leituras do gene *16S rRNA* foram simulados variando os parâmetros do programa *sim16S* de acordo com o descrito na **tabela 3.3**. Os parâmetros variáveis incluíram o número de sequências seleccionadas da base de dados (1000 ou 5000), a proporção de leituras com erros de sequenciação (0%, 1%, 10% ou 100%), o número de mutações por leitura (entre 1 e 4) e o comprimento da leitura (150 pb ou 250 pb conforme o tamanho do respectivo amplicão). O número máximo de erros por leitura foi de 4 por cada 150 bases e de 1 por cada 250 bases, sendo que cada base foi substituída aleatoriamente por uma das restantes 3. A ausência de erros nas leituras foi garantida através da utilização do valor 100000 no parâmetro *mutReads*, para que uma leitura mutada só fosse gerada com uma probabilidade de 1 em 100000.

Tabela 3.3. Valores dos parâmetros de entrada dos conjuntos de dados simulados com o programa *sim16S*.

#	Conjunto de dados	Amplicão	Nº de sequências seleccionadas (*numSeq*)	Comprimento da leitura (*readLen*)	Número de leituras com erros (*mutReads*) [a]	Número de erros por leitura (*numMut*)
1	A_1k_150_noMutations	A	1000	150	<1 (100000)	1
2	B_1k_250_noMutations	B	1000	250	<1 (100000)	1
3	A_1k_150_1%_1_rep1	A	1000	150	~100 (100)	1
4	A_1k_150_1%_1_rep2	A	1000	150	~100 (100)	1
5	A_1k_150_10%_1_rep1	A	1000	150	~1000 (10)	1
6	A_1k_150_10%_1_rep2	A	1000	150	~1000 (10)	1
7	A_1k_150_100%_1_rep1	A	1000	150	10000 (1)	1
8	A_1k_150_100%_1_rep2	A	1000	150	10000 (1)	1
9	A_1k_150_1%_1	A	1000	150	~100 (100)	1
10	A_1k_150_10%_1	A	1000	150	~1000 (10)	1
11	A_1k_150_100%_1	A	1000	150	10000 (1)	1
12	B_1k_250_1%_1	B	1000	250	~100 (100)	1
13	B_1k_250_10%_1	B	1000	250	~1000 (10)	1
14	B_1k_250_100%_1	B	1000	250	10000 (1)	1
15	A_1k_150_10%_2	A	1000	150	~1000 (10)	2
16	A_1k_150_10%_4	A	1000	150	~1000 (10)	4
17	B_5k_250_noMutations	B	5000	250	<1 (100000)	1

(a) Valores estimados de acordo com a probabilidade dada pelo parâmetro *mutReads*.

3.2. Classificação taxonómica de leituras simuladas

A classificação taxonómica das leituras simuladas do gene *16S rRNA* foi efectuada usando os programas *QIIME* (versão 1.9.1) e *mothur* (versão 1.38.1). Os programas foram instalados num computador portátil *Lenovo* com 15.6 GB de memória, processador *Intel© Core™ i7-4510U CPU @ 2.00GHz x4* e sistema operativo *Linux Mint 17.2 Cinnamon 64-bit*. Ambos os programas foram utilizados em modo de linha de comandos. O programa *QIIME* foi utilizado de acordo com o tutorial específico para análise de dados de sequenciação obtidos em plataformas *Illumina* (Illumina Overview Tutorial: Moving Pictures of the Human Microbiome, 2015 que usa o algoritmo *pick_open_reference_otus.py* para agrupar todas as leituras em UTO(s) e não apenas as que têm elevada semelhança com as sequências de uma base de dados de referência. No tutorial não foi utilizado o passo inicial de pré-filtragem de sequências de origem humana e de artefactos de sequenciação, uma vez que o conjunto de dados a analisar foi composto unicamente por sequências dos genes ribossomais. O programa *mothur* foi utilizado de acordo com Schloss *et al.* (2009) e Kozich *et al.* (2013), e com o tutorial descrito em MiSeq SOP (2013), para a plataforma de sequenciação *MiSeq* (*Illumina*), com algumas modificações. Em particular, não foram utilizados os comandos para

construção de *contigs* com base nos ficheiros *fastq* da *read1* e da *read2* produzidos pelo *MiSeq*, detecção de sequências quiméricas e remoção de sequências não pertencentes ao gene *16S rRNA*, uma vez que se partiu de um conjunto de dados simulados. As *pipelines* do *QIIME* e do *mothur* estão esquematizadas na **figura 3.4**.

O primeiro passo do algoritmo *pick_open_reference_otus.py* do *QIIME* consiste na selecção de UTO(s) com o programa *UCLUST* (Edgar, 2010) a um nível de semelhança de 0.97 (e que é efectuada de igual forma nos passos seguintes), usando a base de dados de referência fornecida. As sequências representativas de cada agrupamento são colocadas no ficheiro *step1_rep_set.fna* enquanto as sequências que não têm identidade com a base de dados são colocadas no ficheiro *failures.fasta*. As sequências deste ficheiro são depois amostradas para produzir o ficheiro *subsampled.failures.fna*. No presente trabalho, optou-se por usar a totalidade das sequências para amostragem (parâmetro *-s* com valor *1*) e definiu-se que a amostragem só seria realizada se existissem no mínimo 10 sequências no ficheiro *failures.fasta* (parâmetro *minimum_failure_threshold* com valor *10*), procurando-se desta forma maximizar o número total de sequências a classificar. No segundo passo do algoritmo, as sequências do ficheiro *subsampled.failures.fna* são agrupadas *de novo*, em que o centróide de cada agrupamento é usado como a "nova sequência de referência" no passo seguinte. As sequências representativas das UTO(s) são colocadas no ficheiro *step2_rep_set.fna*. Seguidamente, as sequências do ficheiro *failures.fasta* são cruzadas com a base de dados de sequências dos agrupamentos produzida no segundo passo do algoritmo. As sequências que não têm correspondência nesta base de dados são colocadas no ficheiro *failures_failures.fasta*, a partir do qual são seleccionadas novas UTO(s) para produzir o ficheiro *step4_rep_set.fna*. No passo seguinte, são produzidos os ficheiros com as sequências representativas (*rep_set.fna*) e o mapa de UTO(s) (*final_otu_map.txt*). No presente trabalho, estabeleceu-se que o tamanho mínimo de cada UTO seria de 2 sequências (valor por defeito), pelo que o mapa final (ficheiro *final_otu_map_mc2.txt*) não inclui UTO(s) com apenas 1 sequência atribuída. No passo final, é adicionada a taxonomia a cada uma das sequências contidas no ficheiro *rep_set.fna*, o qual é também usado para fazer o alinhamento múltiplo das sequências com o programa *Pynast* (ficheiro *rep_set_aligned.fasta*) e construir a árvore filogenética (ficheiro *rep_set.tre*). A tabela final de UTO(s) é produzida no formato *biom* (ficheiro *otu_table_mc2_w_tax_no_pynast_failures.biom*).

A tabela de UTO(s), que contém as sequências atribuídas a cada táxon nos vários níveis taxonómicos, pode ser convertida em ficheiro de texto usando o comando *summarize_taxa.py*, em que cada ficheiro gerado corresponde a um nível taxonómico desde o filo ao género. Estes ficheiros podem ser abertos numa folha de cálculo para se obter o total de sequências em cada táxon, conforme descrito no **Anexo D**. No presente trabalho, usaram-se os ficheiros *refSeq.fasta* e *taxonomyRef.txt* produzidos pelo programa *sim16S* como a coleção de sequências de referência e respectiva taxonomia. Neste caso, as leituras do gene *16S rRNA* deverão ter pelo menos 1 alvo naquela colecção, uma vez que são geradas a partir do ficheiro *refSeq.fasta*. O ficheiro *taxonomyRef.txt* criado pelo *sim16S* tem de ser primeiramente configurado usando uma folha de cálculo (**Anexo E**). Adicionalmente, para que o *QIIME* use bases de dados de sequências de referência e de taxonomia personalizadas, como as que foram utilizadas neste trabalho, é necessário criar um ficheiro de configuração conforme indicado no **Anexo F**. Os comandos necessários para executar o *QIIME* estão descritos detalhadamente no **Anexo G**.

O programa *mothur* pode ser executado comando a comando ou de uma só vez em formato de *pipeline*. Neste trabalho, optou-se por dividir a *pipeline* do *mothur* em 6 etapas, conforme descrito no **Anexo H**. O *mothur* também requer a criação de um ficheiro de texto indicando a que grupo (amostra)

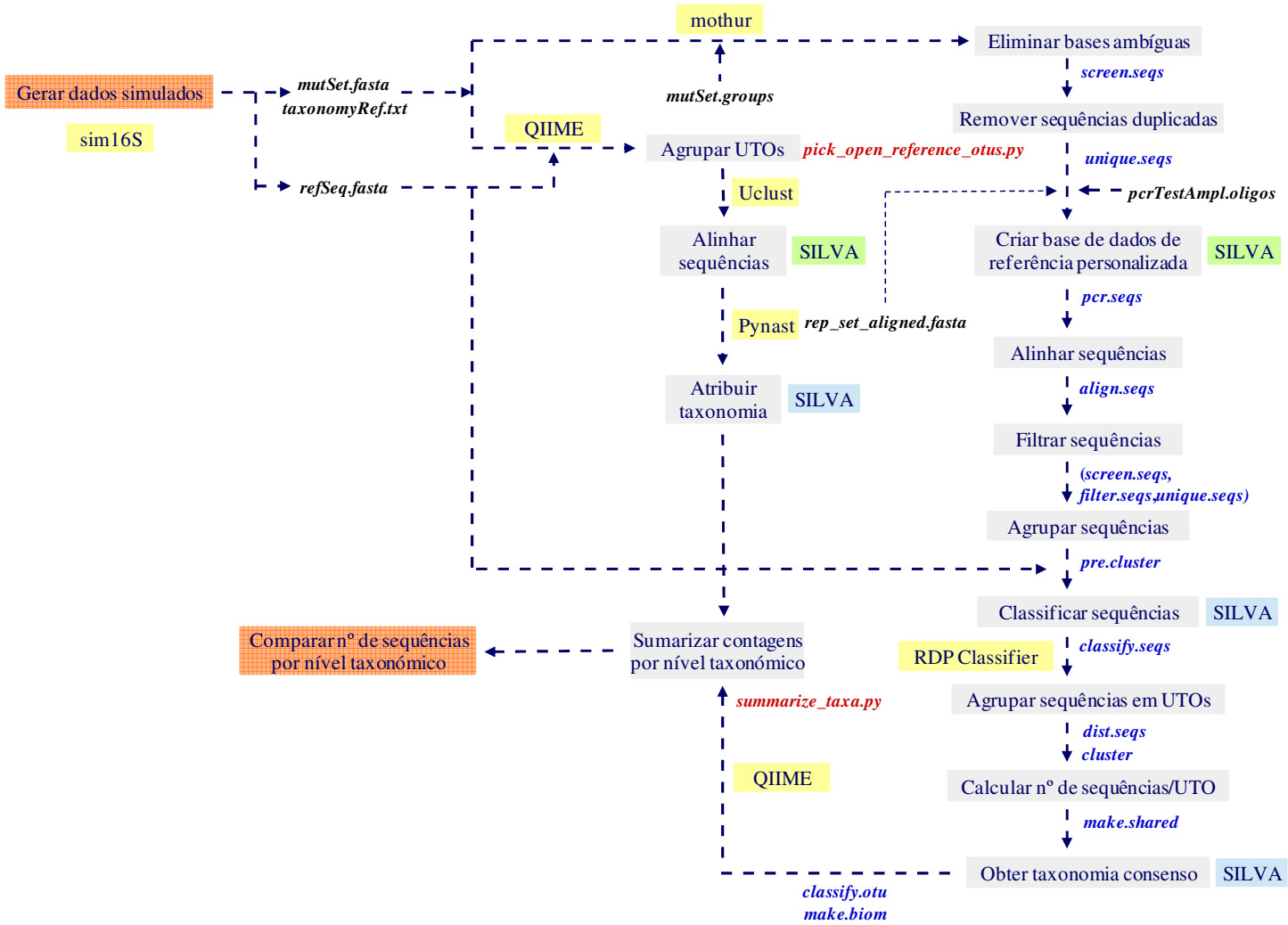

Figura 3.4. Representação esquemática das *pipelines* de análise de dados simulados do gene *16S rRNA* com os programas *QIIME* e *mothur*. As várias fases de análise dos 2 programas iniciam-se com os dados produzidos pelo programa *sim16S* (ficheiro *mutSet.fasta*) e terminam na sumarização das contagens por nível taxonómico, que podem então ser comparadas. As *pipelines* do *mothur* e do *QIIME* estão representadas por linhas contínuas e tracejadas, respectivamente. As linhas a pontiado correspondem a passos comuns aos 2 programas. Os vários programas utilizados estão assinalados com caixas amarelas. As sequências de referência e de taxonomia personalizadas estão representadas por caixas verdes e azuis respectivamente. Os comandos do *QIIME* e do *mothur* estão indicados a letra vermelha e azul, respectivamente.

pertence cada sequência contida no ficheiro *mutSet.fasta* (**Anexo I**). A primeira etapa é iniciada com uma análise do conjunto de dados do ficheiro *mutSet.fasta* através do comando *summary.seqs*, que permite obter estatísticas sobre o comprimento das sequências, número de bases ambíguas e número de homopolímeros. Seguidamente, o comando *screen.seqs* vai remover as leituras que têm 1 ou mais bases ambíguas (parâmetro *maxambig* com valor *0*), produzindo o ficheiro *mutSet.good.fasta*. Uma vez que o conjunto de dados deverá conter muitas sequências iguais entre si, os duplicados são removidos deste ficheiro com o comando *unique.seqs* para evitar posteriormente que a mesma sequência seja alinhada múltiplas vezes, produzindo o ficheiro *mutSet.good.unique.fasta*. Para reduzir o tamanho dos ficheiros (particularmente quando o nome das sequências é muito extenso como acontece nos ficheiros *fastq* criados pelo *MiSeq*), o comando *count.seqs* gera uma tabela em que as linhas são os nomes das sequências únicas e as colunas são o nome dos grupos (neste caso, apenas 1 grupo), a qual é preenchida com o total de leituras que existe para cada sequência e grupo. Seguidamente, antes de se proceder ao alinhamento das leituras únicas, é possível personalizar uma base de dados de sequências de referência que contenha somente a região de interesse (i.e., o amplicão do gene *16S rRNA* gerado pelo *sim16S*), para evitar a análise das sequências génicas completas. Esta base de dados é construída com base num ficheiro de texto contendo a sequência dos 2 oligonucleótidos usados na selecção dos amplicões, conforme descrito no **Anexo J**. O comando *pcr.seqs* vai assim criar a base de dados personalizada usando o ficheiro com as sequências dos oligonucleótidos (*pcrTestAmplA.oligos* ou *pcrTestAmplB.oligos*) e um ficheiro de sequências alinhadas do gene *16S rRNA* que, no presente trabalho, corresponde ao ficheiro *rep_set_aligned.fasta* criado pelo programa *PyNAST* (incluído na *pipeline* do *QIIME*) a partir do mesmo ficheiro *mutSet.fasta* que foi utilizado no comando inicial do *mothur*. No final desta etapa, as leituras são alinhadas com as sequências da base de dados personalizada (ou com outra base de dados de referência incluída originalmente no *mothur*) usando o comando *align.seqs*, através de pesquisa de *k-mers*, produzindo o ficheiro *mutSet.good.unique.align*. O comando *summary.seqs* pode ser novamente corrido para analisar o conjunto de sequências alinhadas (ficheiro *mutSet.good.unique.align*).

O último comando *summary.seqs* permite obter as posições de início e de fim das sequências do ficheiro *mutSet.good.unique.align*. No resultado deste comando, deve procurar-se identificar as posições mais frequentes do alinhamento para usar no comando *screen.seqs* seguinte, o que garante que se irá obter as leituras que cobrem a mesma região do alinhamento, sendo descartadas as leituras que se iniciam após a posição de início e/ou que terminam antes da posição de fim. Esta selecção da região de alinhamento pode ser verificada correndo novamente *summary.seqs* usando o ficheiro *mutSet.good.unique.good.align* produzido pelo comando *screen.seqs*. Seguidamente usa-se o comando *filter.seqs* que permite eliminar os espaços criados pelo alinhamento (por exemplo, caracteres "-"), o que resulta num ficheiro (*mutSet.good.unique.good.filter.fasta*) com um número muito menor de colunas do que o ficheiro inicial. Uma vez que a eliminação de colunas pode ter originado sequências idênticas, através da remoção de sequências nas extremidades, corre-se novamente o comando *unique.seqs* para eliminar sequências duplicadas, o que produz o ficheiro *mutSet.good.unique.good.filter.unique.fasta*. Em seguida, as leituras únicas são pré-agrupadas de acordo com o número de diferenças de bases entre si e que, neste trabalho, foi de 1 base nas leituras de 150 pb (amplicão A) e de 2 bases nas leituras de 250 pb (amplicão B). Após o agrupamento, as sequências do ficheiro *mutSet.good.unique.good.filter.unique.precluster.fasta* podem ser classificadas usando o classificador *Bayesiano* do programa *RDP Classifier*, através da função *classify.seqs*. Esta função usa os ficheiros de sequências de referência (*refSeq.fasta*) e de taxonomia (*taxonomyRef.txt*) produzidos pelo *sim16S*, sendo aquele último configurado conforme descrito no **Anexo E**.

A terceira etapa inicia-se com o agrupamento das sequências em UTO(s) usando o comando

dist.seqs, que calcula as distâncias entre pares de sequências do ficheiro *mutSet.good.unique.good.filter.unique.precluster.fasta* e coloca estes valores numa matriz. Este comando permite definir um *cut-off* para o valor da distância entre sequências que, no presente caso foi de 0.20, ou seja, só são colocados na matriz as distâncias cujo valor seja inferior a 0.20. Com base nesta matriz de distâncias, o comando *cluster* agrupa as sequências em UTO(s) usando o método de agrupamento *OptiClust* (Westcott and Schloss, 2017), que troca as sequências entre grupos taxonómicos por forma a maximizar a qualidade da atribuição. O número total de sequências que existe em cada UTO é dado pelo comando *make.shared*. A taxonomia consenso de cada UTO pode então ser obtida com o comando *classify.otu* usando o ficheiro de taxonomia produzido anteriormente no comando *classify.seqs*. Por fim, o ficheiro *shared* pode ser convertido em formato *biom* com o comando *make.biom* e, usando o programa *QIIME* e o comando *summarize_taxa.py*, obter os dados de taxonomia por nível taxonómico. Desta forma, é possível comparar os dados obtidos no *mothur* com os obtidos no *QIIME* usando o mesmo formato de apresentação.

4. Resultados

4.1. Implementação de um programa em *Matlab* para simulação de leituras do gene *16S rRNA*

A linguagem *Matlab* é baseada em matrizes e constitui uma das linguagens de programação mais utilizadas para responder a problemas técnicos e científicos. Neste trabalho utilizou-se a linguagem *Matlab* para criar um programa (*sim16S*) que produz leituras simuladas do gene *16S rRNA*, a partir de uma base de dados de sequências de referência. As funções do programa *sim16S* foram previamente testadas individualmente e em conjunto para garantir que produziam os resultados expectáveis. Concretamente, foram confirmados os seguintes dados e resultados (não apresentados):

- comprimento das sequências concatenadas e total de sequência extraído da base de dados;
- conversão de bases U em bases T;
- as sequências dos oligonucleótidos correspondem à sequência das extremidades dos amplicões e as sequências excluídas não contêm as sequências dos oligonucleótidos;
- estatísticas dos amplicões;
- contagem de sequências duplicadas de amplicões;
- amplicões seleccionados (após a distribuição de *Poisson*) e respectiva taxonomia estão correctos;
- comprimento das leituras;
- proporção de leituras com erros de sequenciação e com diferente número de erros por leitura;
- distribuição meia-normal dos erros nas leituras;
- contagens de táxones por nível taxonómico;
- dados do relatório.

O tempo total de execução do programa *sim16S* varia principalmente em função do número de sequências que são extraídas da base de dados, da proporção de leituras com erros e do número de erros por leitura. Por exemplo, o tempo de execução foi de aproximadamente 5 segundos para extrair 1000 sequências da base de dados e produzir 10000 leituras aleatórias de 150 pb da região hipervariável V3 contendo 1% de leituras com 1 erro de sequenciação (**figura 4.1**). No caso da extração de 5000 leituras, o tempo de computação necessário para produzir o mesmo número de leituras aleatórias de 250 pb da região hipervariável V4, contendo 10% de leituras mutadas, cada uma com 2 erros de sequenciação, foi de aproximadamente 23 segundos. De forma a confirmar que os oligonucleótidos utilizados permitiam obter os amplicões esperados, compararam-se os resultados do *sim16S* com a informação existente na literatura. Assim, para o par de oligonucleótidos *S-D-Arch-0519-a-S-15/S-D-Bact-0785-b-A-18*, o tamanho médio dos amplicões foi de 284 pb (intervalo: 269-301 pb), o que coincide com o tamanho esperado (284 pb) de acordo com Klindworth *et al.* (2013). Para o par *S-D-Bact-0341-b-S-17/S-D-Bact-0515-a-A-19*, o tamanho médio dos amplicões seleccionados foi de 182 pb (intervalo: 157-197 pb), o que está próximo do tamanho esperado (193 pb) segundo Klindworth *et al.* (2013). No entanto, é importante referir que aqueles conjuntos de dados representam apenas uma pequena fracção do total de sequências presentes na base de dados de referência. A análise do ficheiro *speciesSet.txt* do programa *sim16S* mostrou que, para qualquer dos conjuntos de dados, as sequências dos amplicões seleccionados correspondiam exclusivamente a sequências de táxones pertencentes ao domínio *Bacteria*. Neste trabalho foram produzidos 17 conjuntos distintos de leituras variando os diversos parâmetros do programa *sim16S* (**tabela 4.1**).

A

```
< sim16S Report >

total run time (in seconds): 4.986

Input data:

 reference 16S rRNA database: SILVA_123_SSURef_Nr99_tax_silva.fasta
 forward primer sequence: CCTACGGGAGGCAGCAG
 reverse primer sequence: TTACCGCGGCTGCTGGCAC
 number of sequences screened: 1000
 maximum read length: 150
 number of predicted mutated reads: 100
 number of mutations per read: 1

Output data:

 total number of bases processed: 1606310
 number of amplicons selected: 337
 percentage of primers on target: 33.7
 maximum amplicon length: 197
 minimum amplicon length: 157
 average amplicon length: 182
 number of unique amplicon sequences: 253
 percentage of redundant sequences: 24.9258
 number of taxa in final dataset: 90
 number of mutated reads: 92
 overall mutation percentage: 0.0061333
```

B

```
< sim16S Report >

total run time (in seconds): 22.5905

Input data:

 reference 16S rRNA database: SILVA_123_SSURef_Nr99_tax_silva.fasta
 forward primer sequence: CAGCAGCCGCGGTAA
 reverse primer sequence: TACCAGGGTATCTAATCC
 number of sequences screened: 5000
 maximum read length: 250
 number of predicted mutated reads: 1000
 number of mutations per read: 2

Output data:

 total number of bases processed: 7712547
 number of amplicons selected: 2501
 percentage of primers on target: 50.02
 maximum amplicon length: 301
 minimum amplicon length: 269
 average amplicon length: 284
 number of unique amplicon sequences: 1447
 percentage of redundant sequences: 42.1431
 number of taxa in final dataset: 232
 number of mutated reads: 1051
 overall mutation percentage: 0.08408
```

Figura 4.1. Exemplos de 2 relatórios produzidos pelo programa *sim16S* para conjuntos de leituras simuladas das regiões hipervariáveis V3 (A) e V4 (B) do gene *16S rRNA*. O relatório está dividido em dados de entrada (introduzidos pelo utilizador) e dados de saída gerados pelo programa. No exemplo A, a base de dados *SILVA* (versão 123) foi utilizada para extrair 1000 sequências e produzir 10000 leituras de 150 pb contendo aproximadamente 100 leituras com 1 erro de sequenciação. No exemplo B, a mesma base de dados foi usada para obter 5000 sequências, a partir das quais foram geradas 10000 leituras de 250 pb em que aproximadamente 1000 destas contêm 2 erros de sequenciação. Os dados de saída incluem o total de bases lidas da base de dados, número de amplicões seleccionados, percentagem de oligonucleótidos com sequências alvo na base de dados, comprimento máximo, mínimo e médio dos amplicões, número de amplicões com sequências únicas, percentagem de amplicões com 2 ou mais sequências idênticas no conjunto de amplicões, número de táxones no conjunto final de dados (após randomização), número real de leituras com erros e percentagem global de erros do conjunto de dados. Os valores destes dois últimos dados podem não corresponder exactamente aos dados introduzidos pelo utilizador, uma vez que resultam de uma selecção aleatória de leituras definida com base numa probabilidade. Nos exemplos A e B, a probabilidade de uma leitura conter erros de sequenciação é de 100/10000 (1%) e de 1000/10000 (10%), respectivamente.

Tabela 4.1. Estatísticas principais dos conjuntos de leituras simuladas produzidos pelo programa *sim16S*.

#	Conjunto de dados	Tempo de execução (segundos)	Número total de amplicões	Número de amplicões distintos	Número total de taxa (a)	Número de leituras com erros	Percentagem global de erros
1	A_1k_150_noMutations	3.6455	337	253	91	0	0
2	B_1k_250_noMutations	3.8653	332	275	88	0	0
3	A_1k_150_1%_1_rep1	(b)	337	253	91	118	0.00786
4	A_1k_150_1%_1_rep2	(b)	337	253	91	120	0.008
5	A_1k_150_10%_1_rep1	(b)	337	253	91	1073	0.07153
6	A_1k_150_10%_1_rep2	(b)	337	253	91	978	0.0652
7	A_1k_150_100%_1_rep1	(b)	337	253	91	10000	0.66667
8	A_1k_150_100%_1_rep2	(b)	337	253	91	10000	0.66667
9	A_1k_150_1%_1	5.1425	337	253	91	90	0.006
10	A_1k_150_10%_1	5.3127	337	253	90	989	0.065933
11	A_1k_150_100%_1	6.2626	337	253	91	10000	0.66667
12	B_1k_250_1%_1	5.1653	332	275	91	87	0.00348
13	B_1k_250_10%_1	3.8328	332	275	92	1043	0.04172
14	B_1k_250_100%_1	5.9153	332	275	86	10000	0.4
15	A_1k_150_10%_2	(b)	337	253	91	1034	0.13787
16	A_1k_150_10%_4	(b)	337	253	91	1040	0.2773
17	B_5k_250_noMutations	24.0973	2501	1447	235	0	0

(a) O número total de táxones pode incluir sequências da base de dados com a mesma taxonomia; (b) o tempo total de execução não foi contabilizado uma vez que os conjuntos de dados foram produzidos apenas com a função *mutAmp.m*.

4.2. Classificação taxonómica de leituras sem erros de sequenciação

Os programas *QIIME* e *mothur* foram usados para classificar taxonomicamente as leituras de cada um dos conjuntos de dados produzidos pelo programa *sim16S*. Além dos 17 conjuntos atrás descritos, o *QIIME* e o *mothur* foram também utilizados para re-classificar os conjuntos #9, #10 e #11, usando as bases de dados de sequências de referência e de taxonomia existentes à partida nos pacotes dos respectivos programas (conjuntos de dados #18 a #20). As estatísticas principais da análise dos 20 conjuntos de dados encontram-se descritas na **tabela 4.2**. O programa *sim16S* foi primeiramente utilizado para gerar um conjunto de leituras sem erros dos amplicões A e B (conjuntos de dados #1 e #2, respectivamente), de forma a testar se os programas *QIIME* e *mothur* classificavam correctamente todas as leituras simuladas. Para tal, usou-se uma base de dados de sequências de referência "personalizada" constituída pelas primeiras 1000 sequências da base de dados *SILVA* (versão 123), a partir da qual foram também geradas as leituras simuladas. Desta forma, os programas *QIIME* e *mothur* utilizaram como sequências de referência o conjunto de sequências que foram seleccionadas pelo *sim16S*.

Globalmente, no que respeita ao amplicão A, o *QIIME* classificou um maior número de leituras do que o *mothur*, mas identificou um menor número de táxones do que este, sendo a diferença maior nos níveis taxonómicos inferiores (**figura 4.2**). Por exemplo, ao nível de género, o *mothur* identificou 47 (84%) e o *QIIME* apenas 33 (59%) dos 56 táxones totais gerados pelo *sim16S*. Ambos os programas apresentaram um pequeno conjunto de táxones, nos vários níveis taxonómicos, com um número de leituras inferior ao esperado. Além disso, mostraram uma tendência para apresentar um maior número de leituras não classificadas nos níveis taxonómicos inferiores, particularmente de família e de género. As principais diferenças entre os programas foram verificadas a nível da identificação de novos táxones e na sobre-estimação do número de leituras ao nível do género. Enquanto as leituras atribuídas em excesso a 1 ou mais táxones foram em número muito superior no *mothur* (n=1176, sendo todas atribuídas ao género *Microbispora*) do que no *QIIME* (n=669, correspondentes a

Tabela 4.2. Estatísticas principais dos resultados da classificação taxonómica realizada pelos programas *QIIME* e *mothur* com base em 20 conjuntos de dados do *sim16S*.

#	Conjunto de dados do *sim16S*	Base de dados de referência		Total de leituras classificadas		Total de taxa														
						Filo			Classe			Ordem			Família			Género		
		Q	m	Q	m	S	Q	m	S	Q	m	S	Q	m	S	Q	m	S	Q	m
1	A_1k_150_noMutations	*SILVA_1K*	*SILVA*_1k	9994	9672	8	8	8	15	15	15	36	30	35	42	39	45	56	48	56
2	B_1k_250_noMutations	*SILVA_1K*	*SILVA_1K*	9999	8919	6	6	6	12	12	12	29	28	28	39	37	37	55	50	50
3	A_1k_150_1%_1_rep1	*SILVA_1K*	*SILVA_1K*	9993	9672	8	8	8	15	15	15	36	30	35	42	39	45	56	48	56
4	A_1k_150_1%_1_rep2	*SILVA_1K*	*SILVA_1K*	9994	9672	8	8	8	15	15	15	36	30	35	42	39	45	56	48	57
5	A_1k_150_10%_1_rep1	*SILVA_1K*	*SILVA_1K*	9993	9672	8	8	8	15	15	15	36	30	36	42	39	46	56	48	58
6	A_1k_150_10%_1_rep2	*SILVA_1K*	*SILVA_1K*	9990	9672	8	8	8	15	15	15	36	30	35	42	39	47	56	48	59
7	A_1k_150_100%_1_rep1	*SILVA_1K*	*SILVA_1K*	9987	9672	8	8	8	15	15	15	36	30	34	42	39	45	56	48	57
8	A_1k_150_100%_1_rep2	*SILVA_1K*	*SILVA_1K*	9988	9672	8	8	8	15	15	15	36	30	35	42	39	46	56	48	58
9	A_1k_150_1%_1	*SILVA_1K*	*SILVA_1K*	9998	9658	8	8	8	15	15	15	34	31	33	41	42	44	56	51	55
10	A_1k_150_10%_1	*SILVA_1K*	*SILVA_1K*	9989	9662	8	8	8	15	15	15	35	29	35	41	38	45	55	47	57
11	A_1k_150_100%_1	*SILVA_1K*	*SILVA_1K*	9993	9681	8	8	8	15	15	15	32	31	32	41	43	44	57	54	59
12	B_1k_250_1%_1	*SILVA_1K*	*SILVA_1K*	9994	8990	6	6	6	12	12	12	28	27	27	41	36	39	58	49	53
13	B_1k_250_10%_1	*SILVA_1K*	*SILVA_1K*	9988	8933	6	6	6	12	12	12	29	27	28	42	36	40	58	49	52
14	B_1k_250_100%_1	*SILVA_1K*	*SILVA_1K*	9991	9029	6	6	6	12	12	12	27	27	26	39	36	37	53	47	47
15	A_1k_150_10%_2	*SILVA_1K*	*SILVA_1K*	9992	9672	8	8	8	15	15	15	36	30	36	42	39	49	56	48	64
16	A_1k_150_10%_4	*SILVA_1K*	*SILVA_1K*	9984	9672	8	8	8	15	15	16	36	30	40	42	39	55	56	49	75
17	B_5k_250_noMutations	*SILVA_5k*	*SILVA_5k*	9986	9879	7	7	7	16	16	16	44	39	43	60	57	62	75	75	86
18	A_1k_150_1%_1	*Greengenes*	*SILVA*	9992	9658	8	8	7	15	15	17	34	28	31	41	47	40	56	57	55
19	A_1k_150_10%_1	*Greengenes*	*SILVA*	9951	9662	8	8	7	15	15	16	35	29	32	41	44	41	55	53	54
20	A_1k_150_100%_1	*Greengenes*	*SILVA*	9881	9681	8	8	7	15	15	18	32	29	34	41	47	46	57	67	68

Abreviaturas: m, *mothur*; Q, *QIIME*; S, *sim16S*.

vários géneros incluindo *Microbispora*), este programa classificou 289 leituras nos géneros *Paracoccus* e *Hafnia*, cujas sequências não estavam presentes no conjunto de leituras simuladas.

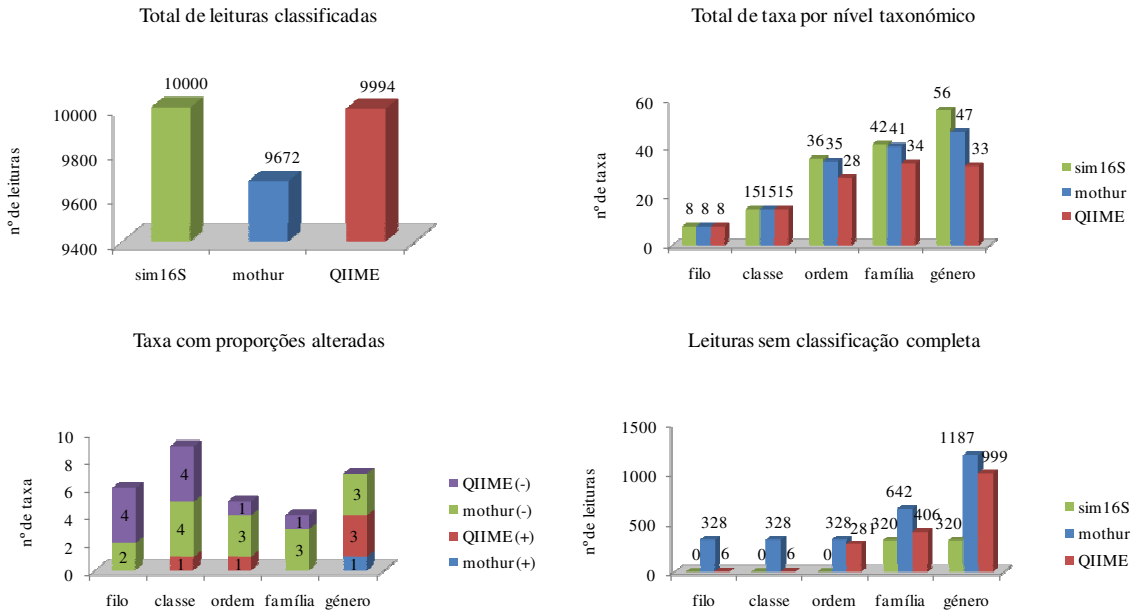

Figura 4.2. Representação gráfica dos resultados de classificação taxonómica produzida pelos programas *QIIME* e *mothur*, para um conjunto de 10000 leituras simuladas do amplicão A sem erros de sequenciação. Nos gráficos "Total de táxones por nível taxonómico" e "Leituras sem classificação completa", os valores apresentados pelo *QIIME* e *mothur* reportam-se aos dados gerados pelo programa *sim16S*. Os sinais (+) e (-) na legenda do gráfico "Taxa com proporções alteradas" indicam os táxones identificados pelos programas que apresentaram um número de leituras maior e menor do que o esperado, respectivamente.

No que respeita ao amplicão B, o *QIIME* classificou um número muito superior de leituras comparativamente ao *mothur*, que rejeitou mais de 10% das leituras simuladas (**figura 4.3**). À semelhança do amplicão A, o *mothur* identificou um maior número de táxones do que o *QIIME*, tendo identificado 50 (91%) dos 55 géneros gerados pelo *sim16S*. No *mothur*, as leituras sem classificação taxonómica envolveram os 5 níveis taxonómicos, enquanto no *QIIME* apresentaram maior expressão nos níveis taxonómicos inferiores. Ao contrário do amplicão A, o *mothur* não mostrou sobre-classificação de leituras em qualquer dos níveis taxonómicos, enquanto o *QIIME* voltou a apresentar leituras por excesso em alguns táxones dos níveis de ordem, família e género. Adicionalmente, o *QIIME* tornou a classificar leituras em géneros que não estavam presentes no conjunto de leituras simuladas, incluindo os géneros *Comamonas* (n=2) e *Serratia* (n=165). Em resumo, a classificação taxonómica de leituras simuladas das regiões hipervariáveis V3 e V4, realizada pelos programas *mothur* e *QIIME* com base em conjuntos de leituras sem erros de sequenciação, aproximou-se globalmente da composição e proporção dos diferentes táxones existentes nos conjuntos de dados simulados. No entanto, o *QIIME* não foi capaz de identificar tantos táxones como o *mothur*, apesar de ter classificado um maior número de leituras. No caso da região V4, a classificação de um maior número de táxones pelo *mothur* foi acompanhado de uma sub-estimação das abundâncias relativas de múltiplos táxones.

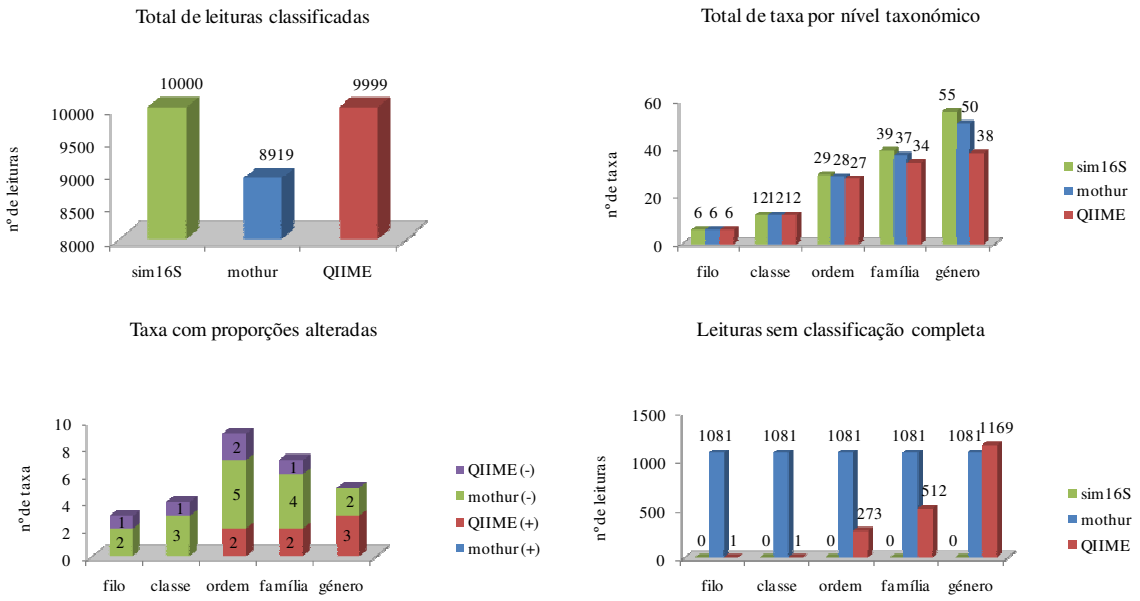

Figura 4.3. Representação gráfica dos resultados de classificação taxonómica produzida pelos programas *QIIME* e *mothur*, para um conjunto de 10000 leituras simuladas do amplicão B sem erros de sequenciação. Nos gráficos "Total de táxones por nível taxonómico" e "Leituras sem classificação completa", os valores apresentados pelo *QIIME* e *mothur* reportam-se aos dados gerados pelo programa *sim16S*. Os sinais (+) e (-) na legenda do gráfico "Taxa com proporções alteradas" indicam os táxones identificados pelos programas que apresentaram um número de leituras maior e menor do que o esperado, respectivamente.

4.3. Classificação taxonómica de leituras com 1 erro de sequenciação

A performance de classificação dos programas *mothur* e *QIIME* foi seguidamente posta à prova utilizando conjuntos de leituras do amplicão A, contendo uma proporção variável de leituras com erros de sequenciação (conjuntos de dados #3 a #8). De forma a poder avaliar-se o efeito dos erros de sequenciação na classificação das leituras, usou-se como ponto de partida o ficheiro *readSet.txt* gerado pela função *readLength.m* em *Matlab*, que serviu de base à criação do conjunto de dados sem erros do amplicão A (ver secção 4.2). Os dados deste ficheiro (10000 leituras simuladas sem erros de sequenciação e com comprimento máximo de 150 pb) foram usados para correr a função *mutAmp.m*, a partir da qual foram gerados conjuntos de dados contendo aproximadamente 1%, aproximadamente 10% e 100% de leituras com 1 erro de sequenciação por leitura. Para obviar os potenciais enviesamentos dos resultados devido ao método de selecção aleatória de leituras com erros, foram gerados 2 replicados de cada conjunto de dados com a mesma proporção teórica de leituras com erros.

Os conjuntos de dados foram classificados taxonomicamente com o *mothur* e o *QIIME* de igual forma ao descrito para os conjuntos de dados sem erros. O *QIIME* produziu um total de táxones dos vários níveis taxonómicos igual ao dos dados sem erros de sequenciação, independentemente da proporção de leituras com erros (resultados não apresentados). Pelo contrário, o *mothur* apresentou uma ligeira tendência para aumentar o número de táxones, em particular ao nível do género, em função da proporção de leituras com erros (**figura 4.4**). No que respeita aos táxones classificados pelo *QIIME*, apenas 1 táxon adicional (em cada replicado dos conjuntos de 100% de leituras com erros) não foi completamente classificado ao nível do género (resultados não apresentados), enquanto no *mothur* o total de táxones sem classificação completa variou entre 1 ao nível da ordem e um máximo de 9 ao nível do género (nos conjuntos de 10% e 100% de leituras com erros), tendo apresentado uma

tendência para aumentar de acordo com uma maior proporção de leituras com erros.

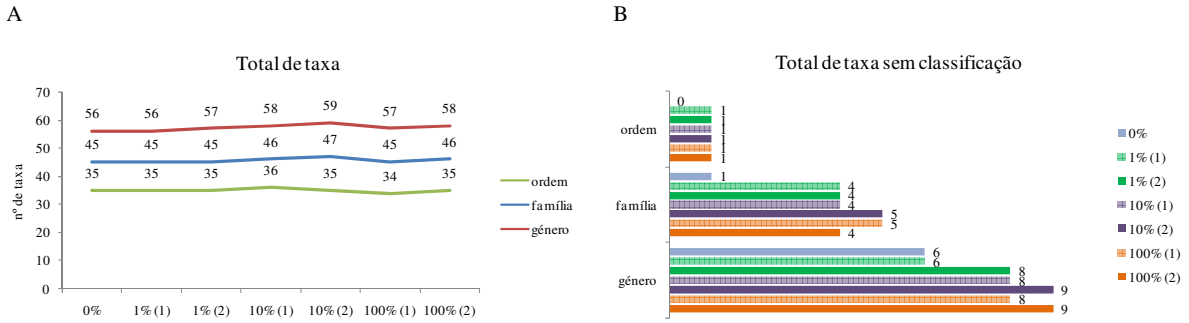

Figura 4.4. Representação gráfica do total de taxa (A) e do total de taxa sem classificação (B) obtida pelo programa *mothur*, ao nível da ordem, família e género, para conjuntos de 10000 leituras simuladas do amplicão A com ~1%, ~10% e 100% de leituras com 1 erro de sequenciação por leitura. Os totais referentes à categoria de 0% são os obtidos com o conjunto de dados sem erros de sequenciação. Os números indicados entre parêntesis após os valores percentuais referem-se aos 2 replicados de cada conjunto de dados analisado.

As classificações produzidas por ambos os programas para os conjuntos de leituras com erros de sequenciação, foram também analisados quanto ao número de leituras atribuídas a cada táxon e à respectiva taxonomia, comparativamente ao conjunto de dados sem erros. Em concreto, para cada nível taxonómico, foi somado o número de leituras sem classificação, o número de leituras atribuídas por excesso (ou defeito) a um táxon já existente (i.e., leituras classifcadas incorrectamente) e o número de leituras atribuídas a táxones que não constavam da classificação dos conjuntos de leituras sem erros, ou seja, foi obtido o total de leituras não classificadas/mal classificadas. Globalmente, o *mothur* apresentou melhor performance de classificação que o *QIIME* ao nível de filo, classe e ordem, mas produziu resultados muito piores nos níveis taxonómicos inferiores (**figura 4.5**). No caso do *QIIME*, a presença de erros afectou maioritariamente a classificação dos géneros *Microbispora* e *Nonomuraea*, ambos pertencentes à família *Streptosporangiaceae*. Em particular, num dos replicados com 100% de leituras com erros, o *QIIME* classificou todas as leituras (n=562) do género *Microbispora* no género *Nonomuraea*, enquanto no outro replicado atribuiu todas estas leituras a um táxon não classificado da família *Streptosporangiaceae*. Além disso, neste último replicado, o *QIIME* classificou todas as leituras (n=59) da família *Nocardiaceae* na família *Tsukamurellaceae* (e consequentemente no género *Tsukamurella*), a qual não constava da classificação do conjunto de leituras sem erros de sequenciação. Contrariamente ao *QIIME*, os problemas de classificação do *mothur* verificaram-se maioritariamente ao nível taxonómico de família. Em particular, o *mothur* não conseguiu classificar a maior parte (ou todas) as leituras das famílias *Streptomycetaceae*, *Thermomonosporaceae*, *Phyllobacteriaceae* e/ou *Nocardiaceae*, e dos respectivos géneros, em diferentes replicados dos conjuntos de dados. No entanto, o número de leituras não classificadas/mal classificadas pelo *mothur* não pareceu variar em função da proporção de leituras com erros em cada conjunto de dados, ao contrário do que se verificou com o *QIIME*. Em resumo, a introdução de erros nas leituras afectou de forma mais evidente a classificação efectuada pelo *mothur*, tendo conduzido a uma sobre-estimação do número de táxones originais, a um maior número de táxones sem classificação completa e a um maior número de leituras classificadas incorrectamente. Por outras palavras, os resultados obtidos pelo *QIIME* aproximaram-se mais da classificação de leituras sem erros, quer ao nível da composição de táxones quer ao nível das respetivas proporções, evidenciando assim uma maior resistência deste método de classificação a um baixo nível de erros de sequenciação.

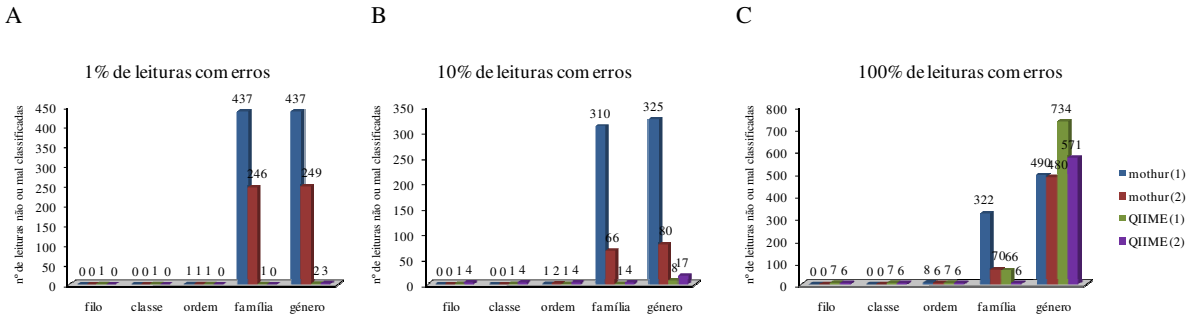

Figura 4.5. Representação gráfica do número de leituras não classificadas/mal classificadas produzida pelos programas *mothur* e *QIIME* para conjuntos de 10000 leituras simuladas do amplicão A com ~1% (A), ~10% (B) e 100% (C) de leituras com 1 erro de sequenciação por leitura. Os números indicados entre parêntesis após os nomes dos programas referem-se aos 2 replicados de cada conjunto de dados analisado.

Os resultados atrás descritos foram obtidos a partir de um único conjunto de dados simulados da região hipervariável V3 do gene *16S rRNA*, no qual foram introduzidas alterações de bases em proporções variáveis. Ou seja, os resultados observados poderiam ser consequência, pelo menos em parte, da composição taxonómica específica do conjunto de dados que foi seleccionado para análise do efeito dos erros. Assim, compararam-se as diferenças de classificação entre o *QIIME* e o *mothur* com base em 3 novos conjuntos de dados independentes dos 2 amplicões, contendo 1 erro por leitura em aproximadamente 1%, aproximadamente 10% ou 100% das leituras (conjuntos de dados #9 a #14). No que respeita aos conjuntos de dados do amplicão A (região hipervariável V3), o *QIIME* classificou um maior número de leituras e identificou um menor número de táxones do que o *mothur* (**figura 4.6**), lembrando os resultados obtidos com leituras sem erros de sequenciação. Ambos os programas tenderam a subestimar o número de leituras de um reduzido conjunto de táxones, mas o *QIIME* apresentou uma tendência para sobre-estimar o número de leituras em alguns táxones, particularmente ao nível do género. O *mothur* apresentou um número superior ao *QIIME* de leituras sem classificação completa, em todos os níveis taxonómicos dos 3 conjuntos de dados, à exceção do género no conjunto com 100% de leituras com erros. Além disso, o *mothur* revelou também um número de leituras aumentado no género *Microbispora*, enquanto no *QIIME* as leituras por excesso foram em menor número e envolveram mais do que 1 género. Este programa voltou também a demonstrar uma tendência para incluir novos táxones na classificação, que incluíram os géneros *Hafnia*, *Paracoccus* e *Tamlana*. Globalmente, a diferente proporção de erros de sequenciação não teve um impacto directamente proporcional no total de leituras classificadas, total de táxones por nível taxonómico ou nas leituras sem classificação completa ou atribuídas por excesso a diferentes táxones.

A classificação de leituras com erros do amplicão B (região hipervariável V4) revelou que apenas cerca de 90% (8933-9029 leituras) das leituras totais produzidas pelo *sim16S* foram classificadas pelo *mothur* (**figura 4.7**). No entanto, o *mothur* identificou um maior número de táxones que o *QIIME*, replicando os resultados obtidos com o amplicão A. O baixo número de leituras classificadas pelo *mothur* reflectiu-se num maior conjunto de táxones com leituras inferior ao esperado e com um maior número de leituras sem classificação completa, comparativamente ao *QIIME*. Em contraste com o amplicão A, o *mothur* não sobre-estimou o número de leituras de qualquer táxon, enquanto o *QIIME* sobre-valorizou o número de leituras de 2 ou mais táxones nos níveis de ordem, família e género. À semelhança do amplicão A, o *QIIME* classificou leituras em táxones não presentes no conjunto de dados do *sim16S*, incluindo os géneros *Serratia*, *Tamlana* e *Comamonas*. Globalmente, os resultados do amplicão B foram semelhantes aos do amplicão A, indicando que diferentes proporções de erros podem não ter impactos distintos conforme as sequências seleccionadas em cada conjunto de dados.

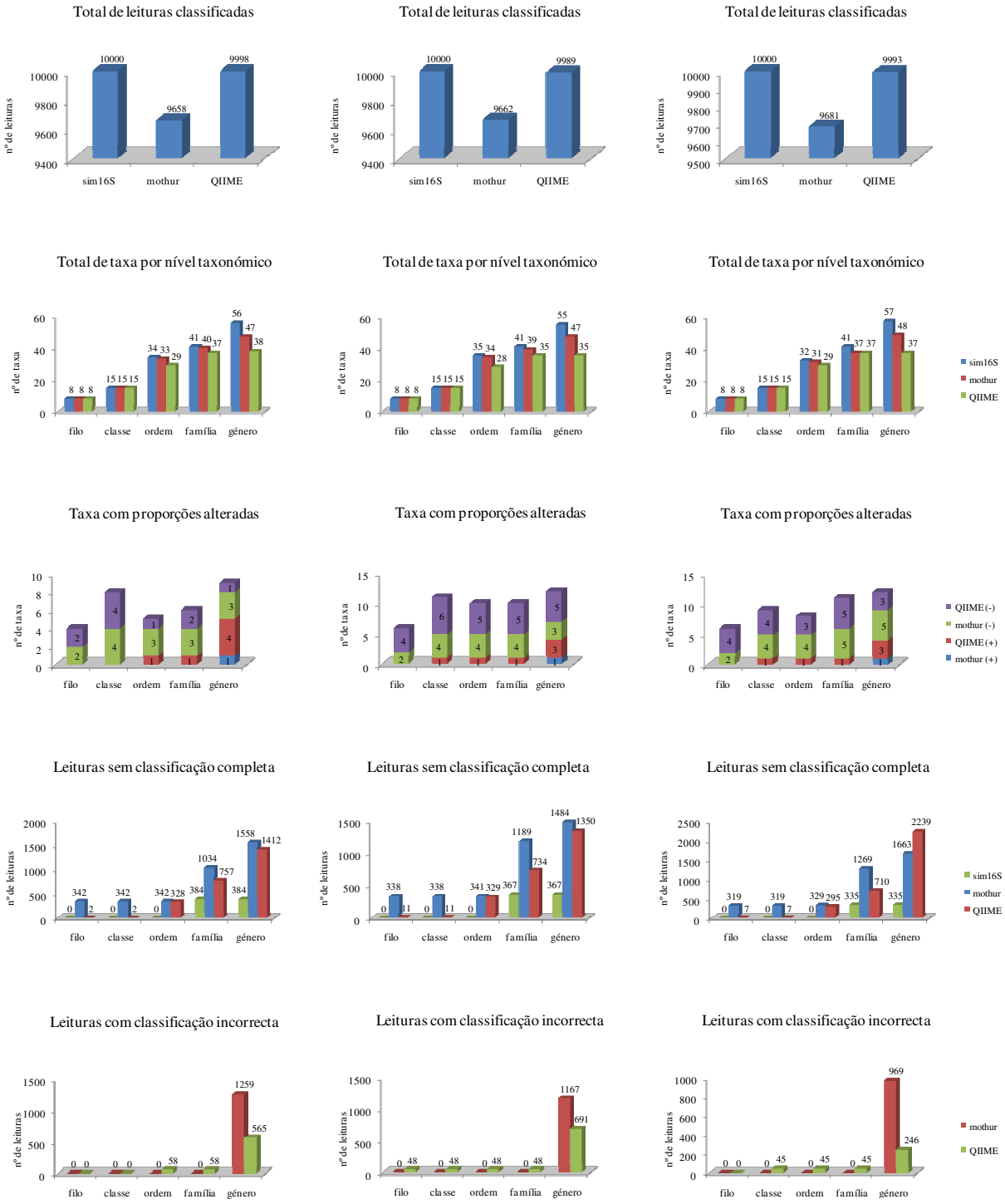

Figura 4.6. Representação gráfica dos resultados de classificação taxonómica produzida pelos programas *QIIME* e *mothur* para conjuntos de 10000 leituras do amplicão A (região hipervariável V3), em que a proporção de leituras com 1 erro de sequenciação por leitura foi de ~1% (coluna da esquerda), ~10% (coluna do centro) e 100% (coluna da direita). No gráfico "Total de taxa por nível taxonómico" os valores apresentados pelo *QIIME* e *mothur* reportam-se ao número de táxones gerados pelo programa *sim16S*. As leituras com classificação incorrecta referem-se a leituras atribuídas por excesso a um ou mais táxones, relativamente ao número de leituras esperado.

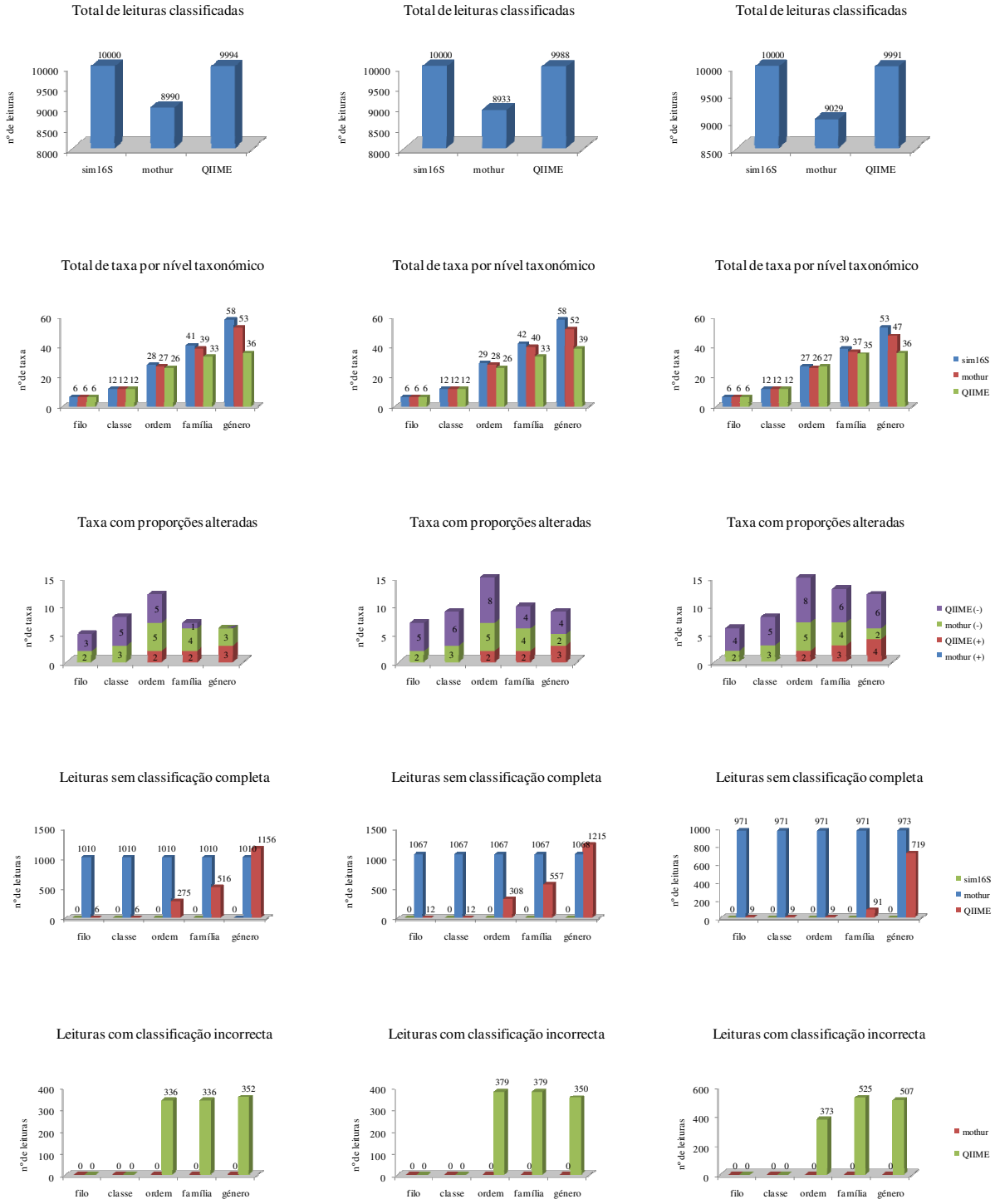

Figura 4.7. Representação gráfica dos resultados de classificação taxonómica produzida pelos programas *QIIME* e *mothur* para conjuntos de 10000 leituras do amplicão B (região hipervariável V4), em que a proporção de leituras com 1 erro de sequenciação por leitura foi de ~1% (coluna da esquerda), ~10% (coluna do centro) e 100% (coluna da direita). No gráfico "Total de taxa por nível taxonómico" os valores apresentados pelo *QIIME* e *mothur* reportam-se ao número de táxones gerados pelo programa *sim16S*. As leituras com classificação incorrecta referem-se a leituras atribuídas por excesso a um ou mais táxones, relativamente ao número de leituras esperado.

4.4. Efeito do número de erros de sequenciação na classificação taxonómica de leituras

Na secção anterior, mostrou-se que a performance de classificação dos programas *mothur* e *QIIME* pode ser afectada quando os conjuntos de dados contêm uma proporção variável de leituras com 1 erro de sequenciação. No entanto, o *QIIME* apresentou uma maior resistência que o *mothur* na classificação das leituras com erros, relativamente aos correspondentes conjuntos de dados sem erros. Para verificar se estas diferenças se mantêm em presença de leituras com um maior número de erros por leitura, testou-se a classificação de conjuntos de dados do amplicão A contendo aproximadamente 10% de leituras com 2 ou 4 erros de sequenciação (conjuntos de dados #15 e #16). À semelhança da secção 4.3, os dados foram gerados a partir do ficheiro *readSet.txt* correspondente ao conjunto de leituras sem erros do amplicão A. Os resultados foram comparados com a classificação produzida pelo *QIIME* e *mothur* com base nos conjuntos de dados homólogos sem erros (secção 4.2) e/ou com aproximadamente 10% de leituras com 1 erro de sequenciação (secção 4.3). Em particular, foram comparados o número total de táxones identificados e o número total de leituras sem classificação completa entre os vários conjuntos de dados (**figura 4.8**). O *mothur* confirmou a tendência para sobre-estimar o número de táxones dos vários níveis taxonómicos (à exceção do filo) em função do número de erros das leituras, enquanto o *QIIME* manteve o número de táxones praticamente constante relativamente ao conjunto de dados sem erros. Em particular, o *mothur* classificou as leituras do conjunto de dados com 4 erros por leitura em 75 géneros, ou seja, mais 19 géneros do que foram classificados no conjunto correspondente de dados sem erros, correspondendo a um total de 1253 leituras. Pelo contrário, o *QIIME* colocou apenas mais 60 leituras num género adicional. Estes resultados comprovam que o *mothur* é bastante mais sensível à presença de leituras com erros do que o *QIIME*, o que se traduz na sobre-estimação da riqueza específica do conjunto de microorganismos dos dados simulados.

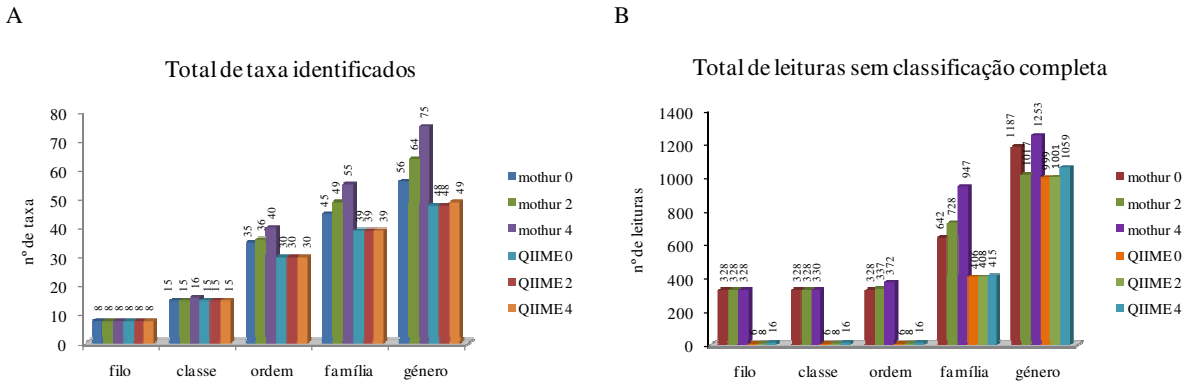

Figura 4.8. Representação gráfica dos resultados de classificação taxonómica produzida pelos programas *QIIME* e *mothur* para conjuntos de 10000 leituras do amplicão A (região hipervariável V3) contendo ~10% de leituras com 2 ou 4 erros por leitura. O gráfico em (A) mostra o total de táxones identificados em cada conjunto de dados com erros, relativamente ao conjunto sem erros (*mothur* 0 e *QIIME* 0), enquanto o gráfico em (B) representa o total de leituras que não foram classificadas completamente para cada conjunto de leituras com erros relativamente ao conjunto de leituras sem erros. Os números à direita dos nomes dos programas nas legendas indicam o número de erros de cada leitura no conjunto de dados.

4.5. Impacto da composição taxonómica dos conjuntos de dados na classificação de leituras

A comparação da performance de classificação dos programas *QIIME* e *mothur*, com base em conjuntos de leituras sem erros de sequenciação, revelou algumas diferenças importantes entre os 2 programas (ver secção 4.1). No entanto, nessa comparação, usou-se unicamente uma colecção de 1000 sequências de referência para gerar os conjuntos de leituras, ou seja, os resultados da comparação poderiam ser influenciados pelo conjunto específico de sequências dos amplicões (e respectivos táxones), seleccionados pelo programa *sim16S*. Assim, estendeu-se a comparação a um conjunto de leituras do amplicão B formado a partir de uma colecção de 5000 sequências de referência (conjunto de dados #17). Tendo em conta o processo de selecção aleatória de sequências do *sim16S*, era de esperar que naquele conjunto fossem incluídos novos táxones com sequências distintas das que foram obtidas com a colecção de 1000 sequências. De facto, a análise comparativa dos táxones existentes nos conjuntos de leituras do amplicão B, gerados a partir de 1000 e 5000 sequências de referência, mostrou que do total de 8 filos seleccionados, apenas 5 estavam presentes em ambos os conjuntos (resultados não apresentados), confirmando que a composição taxonómica é muito diferente nos 2 conjuntos. Devido ao elevado número de táxones gerados a partir de 5000 sequências do gene *16S rRNA*, a comparação destes conjuntos de dados só incluiu os níveis taxonómicos de filo, classe, ordem e família.

O total de leituras classificadas pelo *mothur* no conjunto formado a partir de 5000 sequências, foi muito superior ao obtido com a colecção de 1000 sequências, tendo sido classificado praticamente o mesmo número de leituras que o *QIIME* (**figura 4.9**). À semelhança deste conjunto, o *mothur* detectou

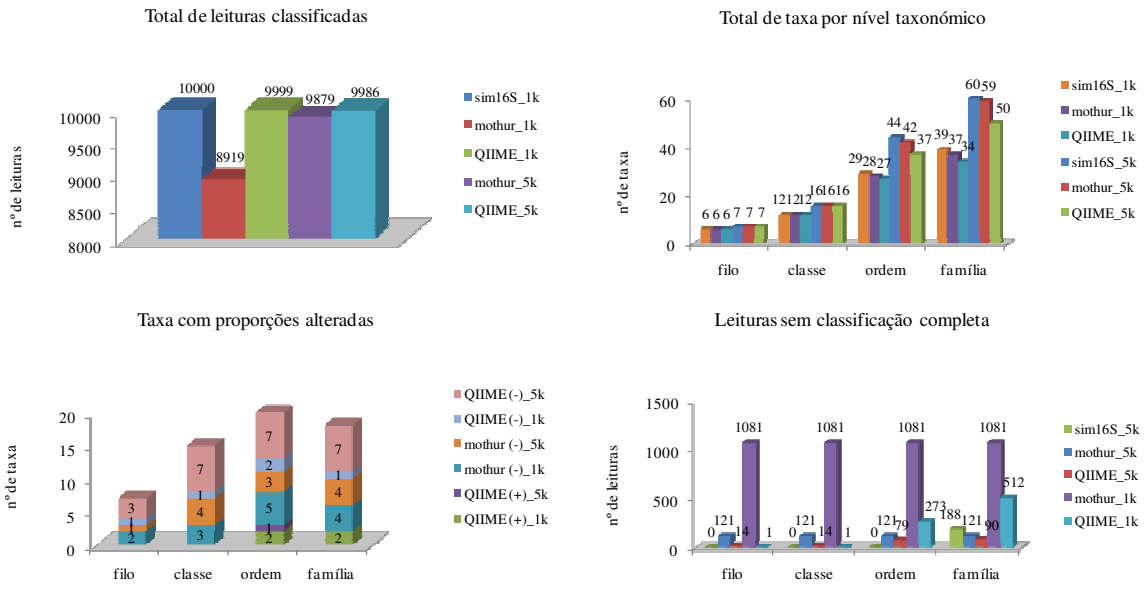

Figura 4.9. Representação gráfica dos resultados de classificação taxonómica produzida pelos programas *QIIME* e *mothur* para conjuntos de 10000 leituras do amplicão B (região hipervariável V4), gerados a partir de colecções contendo 1000 (1k) ou 5000 (5k) sequências de referência. Os resultados do gráfico "Total de taxa por nível taxonómico" incluem o número total de táxones gerados pelo programa *sim16S* para comparação. Os sinais (+) e (-) na legenda do gráfico "Taxa com proporções alteradas" indicam os táxones identificados pelos programas que apresentaram um número de leituras maior e menor do que o esperado, respectivamente.

uma maior proporção dos táxones gerados pelo *sim16S* do que o *QIIME*, nos níveis taxonómicos de ordem e família. Além disso, a diferença máxima entre o número real de táxones produzido pelo *sim16S* e o número de táxones classificados pelo *QIIME*, foi maior no conjunto de dados formado a partir de 5000 sequências (n=10) do que de 1000 sequências (n=5). O *QIIME* também apresentou um maior número de táxones com leituras abaixo do esperado comparativamente ao *mothur*. As leituras não classificadas completamente foram quantitativamente semelhantes em ambos os programas, ao nível da família, o que está em concordância com os resultados obtidos com o conjunto de leituras gerado a partir de 1000 sequências de referência, para o nível de género (ver figura 4.3). No que respeita a novos táxones, apenas o *QIIME* classificou leituras em famílias não incluídas no conjunto de leituras, nomeadamente as famílias *Hyphomicrobiaceae* (n=17) e *Pseudoalteromonadaceae* (n=88). Em resumo, o *mothur* mostrou uma performance superior ao *QIIME* na classificação de leituras com base numa colecção maior de sequências de referência, tendo identificado um maior número dos táxones originais e respectivas proporções no conjunto de dados.

4.6. Influência das bases de dados de sequências do gene *16S rRNA* na classificação taxonómica

A classificação taxonómica dos conjuntos de dados referidos nas secções anteriores baseou-se numa colecção "personalizada" e reduzida de sequências da base de dados *SILVA*, constituída por 1000 ou 5000 sequências. Para avaliar a influência que a escolha da base de dados de referência poderá ter na classificação taxonómica de leituras do gene *16S rRNA*, os conjuntos de leituras simuladas independentes do amplicão A, contendo ~1%, ~10% e 100% de leituras com 1 erro de sequenciação por leitura, foram classificados com as sequências de referência usadas por defeito em cada um dos programas (conjuntos de dados #18 a #20). O *mothur* usa a base de dados *SILVA* (versão 102) enquanto o *QIIME* usa a *Greengenes* (versão 13.8). Nesta comparação, não foram consideradas as leituras atribuídas por defeito ou excesso aos diferentes táxones, nem a identificação de novos táxones, uma vez que a utilização de diferentes taxonomias nas 2 bases de dados dificulta o cruzamento da informação taxonómica. Assim, determinou-se apenas o total de leituras classificadas e a proporção de táxones não classificados em cada nível taxonómico. O número total de leituras classificadas pelo *QIIME* foi superior ao *mothur* em cada um dos conjuntos de dados analisados, embora apresentasse uma tendência para diminuir em função da proporção de leituras com erros, sendo a diferença para o *mothur* inferior à obtida com a versão reduzida da base de dados *SILVA* (**figura 4.10**). No entanto, a proporção de táxones não classificados, isto é, a percentagem do número de táxones sem classificação completa no total de táxones identificados em cada nível taxonómico, foi muito superior à observada anteriormente. Em ambos os programas, a proporção de táxones não classificados aumentou em direcção aos níveis taxonómicos inferiores em qualquer dos conjuntos de dados e mostrou uma tendência para aumentar em função da proporção de leituras com erros. No caso do *mothur*, os táxones não classificados estavam presentes em todos os níveis taxonómicos, enquanto no *QIIME* apenas existiram na família e no género. Curiosamente, a grande diferença entre as proporções ao nível da família entre o *mothur* e o *QIIME* tendeu a esbater-se ao nível do género. Neste conjunto de dados, quase metade (48,5%) dos táxones definidos pelo *mothur* não foram classificados ao nível do género. Este resultado pode dever-se ao elevado número de sequências únicas (n=1569) classificadas pelo *mothur* com a base de dados por defeito, comparativamente com o número de sequências únicas (n=466) classificadas com a base de dados *SILVA* reduzida (**Anexo L**). À semelhança do que foi observado anteriormente (ver secções 4.3 e 4.4), o *QIIME* exibiu uma maior resistência à presença de erros de sequenciação que o *mothur*.

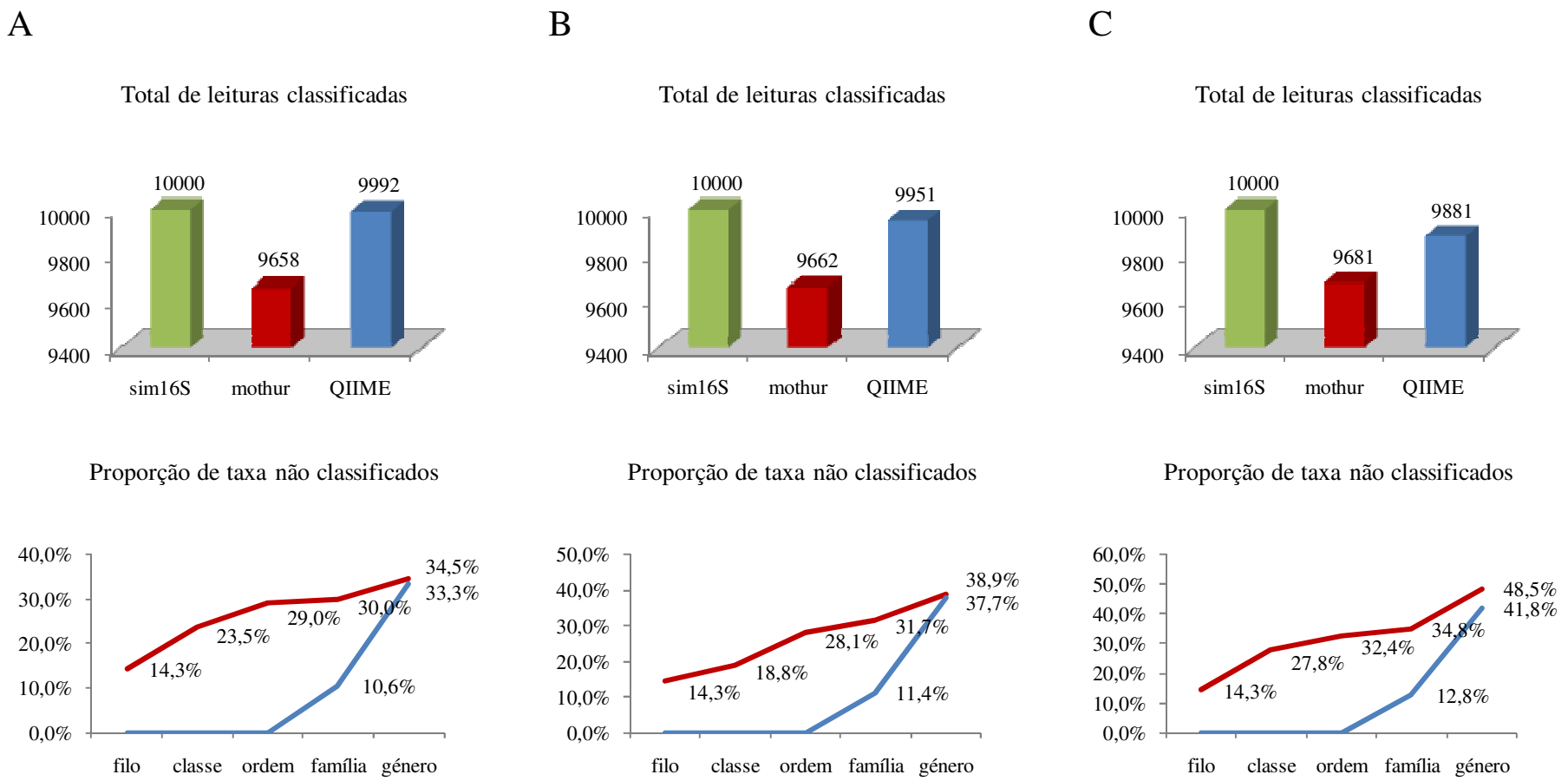

Figura 4.10. Representação gráfica dos resultados de classificação taxonómica produzida pelos programas *QIIME* e *mothur*¸ usando as bases de dados de sequências de referência *SILVA* (*mothur*) e *Greengenes* (*QIIME*), para conjuntos de 10000 leituras do amplicão A (região hipervariável V3) contendo (A) ~1%, (B) ~10% e (C) 100% de leituras com 1 erro de sequenciação por leitura. A proporção de táxones não classificados corresponde à percentagem do número de táxones sem classificação completa, no respectivo nível taxonómico, relativamente ao total de táxones identificados nesse mesmo nível.

4.7. Exactidão da classificação taxonómica

Nas secções anteriores foram analisados os resultados da classificação de diversos conjuntos de dados simulados usando o *QIIME* e o *mothur*. Nomeadamente, a performance de classificação dos 2 programas foi comparada usando conjuntos de leituras contendo proporções variáveis de erros de sequenciação. Os resultados foram analisados de forma detalhada no que respeita ao número total de táxones identificados em cada nível taxonómico e das respectivas proporções relativas, e das leituras sem classificação taxonómica completa ou que foram atribuídas a táxones não existentes no conjunto de dados iniciais. Como se observou anteriormente, os 2 programas mostraram resultados diferentes consoante o nível de análise efectuado, não sendo possível evidenciar facilmente qual dos 2 programas demonstrou a melhor performance global de classificação taxonómica. Neste contexto, calculou-se a exactidão de classificação de ambos os programas como forma de sumarizar num único valor a performance global de classificação. A exactidão foi calculada para cada um dos níveis taxonómicos separadamente subtraindo ao total de leituras (10000) geradas pelo *sim16S*, as leituras excluídas da classificação, as leituras não classificadas completamente, as leituras classificadas em táxones não existentes no conjunto de dados e as leituras seleccionadas pelo *sim16S* cuja taxonomia não se encontrava completa na base de dados *SILVA* para um determinado nível taxonómico. O cálculo da exactidão teve como pressuposto que as leituras classificadas num dado táxon são, efectivamente, as mesmas leituras que originalmente pertenciam a esse táxon, ou seja, a exactidão foi calculada com base em dados agregados e não com base na análise das leituras individuais. No que respeita ao conjunto de leituras sem erros do amplicão A (região hipervariável V3), o *QIIME* apresentou uma melhor exactidão do que o *mothur* para os níveis taxonómicos de filo, classe, família e género, tendo ficado apenas ligeiramente atrás do *mothur* ao nível da ordem (**figura 4.11**). O *QIIME* exibiu também uma exactidão superior ao *mothur* em todos os níveis taxonómicos no conjunto de leituras sem erros do amplicão B (região hipervariável V4), à exceção do género. Ao contrário do *mothur*, o *QIIME* evidenciou uma exactidão mais concordante entre os 2 conjuntos de dados nos 5 níveis taxonómicos analisados, indicando que a performance de classificação não foi afectada pela selecção da região hipervariável do gene *16S rRNA*. Em particular, a exactidão de classificação deste programa foi de praticamente 100% ao nível do filo e da classe para qualquer das regiões hipervariáveis do gene *16S rRNA*.

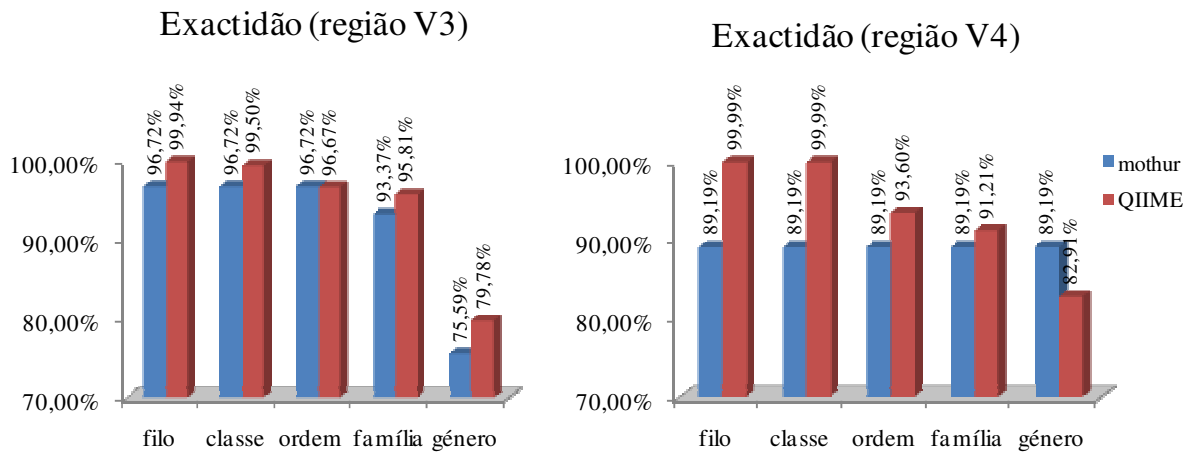

Figura 4.11. Representação gráfica da exactidão da classificação taxonómica dos programas *QIIME* e *mothur* baseada em conjuntos de 10000 leituras sem erros do amplicão A (região hipervariável V3) e do amplicão B (região hipervariável V4).

A exactidão da classificação taxonómica também foi determinada para os conjuntos de dados do amplicão A contendo diferentes percentagens de leituras com erros de sequenciação. Nestes casos, a exactidão correspondeu ao somatório das leituras não classificadas completamente, leituras atribuídas a novos táxones e leituras atribuídas por excesso a táxones existentes, relativamente ao total de leituras classificadas por cada programa no respectivo conjunto de dados sem erros de sequenciação. Por outras palavras, calculou-se em cada conjunto de dados a percentagem média de leituras de ambos os replicados, cuja classificação era igual à classificação obtida nas leituras sem erros. Os resultados mostraram que a exactidão da classificação taxonómica foi superior a 99,9% em qualquer dos 2 programas para os níveis taxonómicos de filo, classe e ordem, independentemente da percentagem de leituras com 1 erro de sequenciação (**figura 4.12**). Pelo contrário, apenas o *QIIME* exibiu uma exactidão superior a 99% para os restantes níveis taxonómicos, nos conjuntos de dados contendo aproximadamente 1% ou 10% de leituras com 1 erro. Ao nível da família e do género, o *mothur* apresentou uma menor exactidão do que o *QIIME* para qualquer dos conjuntos de dados, à exceção da classificação ao nível de género nos dados contendo 100% de leituras com erros.

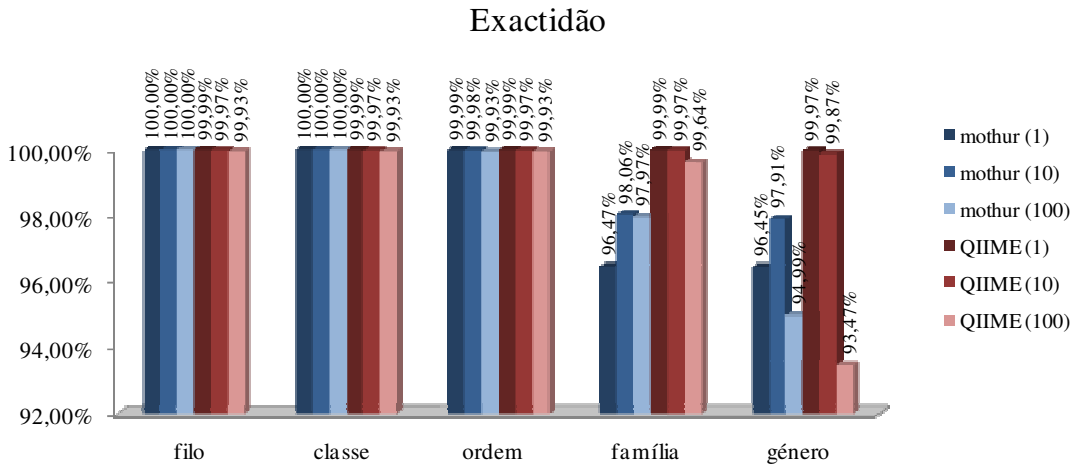

Figura 4.12. Representação gráfica da exactidão da classificação taxonómica dos programas *QIIME* e *mothur* obtida nos conjuntos de dados contendo ~1%, ~10% e 100% de leituras com 1 erro de sequenciação, relativamente à classificação do conjunto de dados sem erros do amplicão A (região hipervariável V3). Os números colocados na legenda entre parêntesis após o nome dos programas correspondem à percentagem de leituras com erros de sequenciação.

Para verificar se a maior exactidão revelada pelo *QIIME* se mantinha em presença de conjuntos de leituras com um maior número de erros, foi calculada a exactidão dos conjuntos de dados contendo aproximadamente 10% de leituras com 2 e 4 erros de sequenciação, previamente classificados com o *mothur* e com o *QIIME* (ver secção 4.4). À semelhança dos conjuntos de dados com apenas 1 erro, a exactidão da classificação foi superior a 99,5% nos níveis taxonómicos de filo, classe e ordem nos conjuntos de leituras com 2 ou 4 erros de sequenciação (**figura 4.13**). Nos restantes níveis taxonómicos, o *QIIME* voltou a exibir uma elevada exactidão da classificação taxonómica (> 99,3%), mesmo em presença de leituras com 4 erros de sequenciação. Adicionalmente, o *QIIME* mostrou apenas uma ligeira redução da exactidão no conjunto de dados com 4 erros relativamente ao conjunto com 2 erros, enquanto no *mothur* essa redução foi comparativamente bastante acentuada. Globalmente, em face dos resultados aqui apresentados, pode afirmar-se que o programa *QIIME* demonstrou uma melhor performance de classificação taxonómica do que o *mothur*, com base em leituras simuladas do gene *16S rRNA* contendo erros de sequenciação.

A análise de exactidão revelou que alguns dos resultados obtidos não estão de acordo com o que seria expectável à partida. No *mothur*, a exactidão de classificação do conjunto de dados com

aproximadamente 1% de leituras com 1 erro por leitura, ao nível da família e do género, foi inferior à dos conjuntos de dados com aproximadamente 10% e 100% (ao nível da família) de leituras com 1 erro por leitura. Adicionalmente, a exactidão do conjunto de dados com aproximadamente 10% de leituras com 2 erros por leitura, foi cerca de 1% superior à do conjunto com apenas 1 erro por leitura. Por outro lado, no *QIIME*, a exactidão do conjunto de dados com 2 erros por leitura foi ligeiramente superior (0,01%) à do conjunto com 1 erro por leitura, nos níveis taxonómicos desde o filo até à família. Estes resultados deverão ser primeiramente interpretados à luz do processo aleatório que o *sim16S* utiliza para introduzir alterações de bases nas leituras, o que implica que as leituras alteradas em cada conjunto de dados não serão muito provavelmente as mesmas. Consequentemente, o alinhamento e/ou agrupamento das leituras no *QIIME* e *mothur* poderão afectar os resultados da classificação taxonómica de cada conjunto de dados. Esta situação foi ilustrada anteriormente quando se analisaram os resultados de 2 replicados do mesmo conjunto de dados inicial, os quais foram sujeitos a processos independentes de introdução de erros nas leituras (ver secção 4.3).

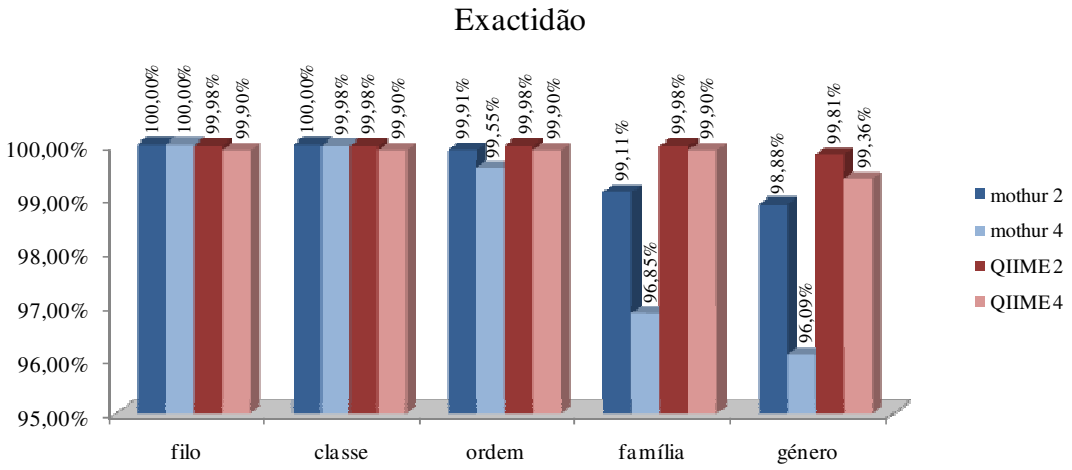

Figura 4.13. Representação gráfica da exactidão da classificação taxonómica dos programas QIIME e mothur obtida nos conjuntos de dados contendo ~10% de leituras com 2 ou 4 erros de sequenciação, relativamente à classificação do conjunto de dados sem erros do amplicão A (região hipervariável V3). Os números colocados na legenda após o nome dos programas correspondem ao número de erros por leitura em cada conjunto de dados.

5. Discussão

A análise bioinformática de leituras de sequenciação do gene *16S rRNA*, produzidas pelas atuais plataformas de sequenciação de nova geração, continua a levantar desafios no que respeita à classificação taxonómica das diferentes espécies de microorganismos em estudos de metagenómica. A sequenciação de amplicões do gene *16S rRNA* produz normalmente um volume de dados relativamente reduzido (cerca de 100000 leituras por amostra em média), comparativamente aos produzidos pela sequenciação *shotgun* (usualmente superior a 1 milhão de leituras por amostra) e de muito menor complexidade do que estes. No entanto, pode ser surpreendente como uma sequência com apenas algumas centenas de pares de bases pode, por vezes, ser tão difícil de associar com uma determinada espécie de microorganismo. Para esta dificuldade contribuem diversos tipos de factores, como sejam as características biológicas da amostra, a metodologia usada para construção das bibliotecas de fragmentos de DNA, os erros inerentes à química da plataforma de sequenciação utilizada, as bases de dados de sequências do gene *16S rRNA*, e os métodos e algoritmos usados para a classificação taxonómica das leituras de classificação.

Um dos problemas inerentes à amostra é a possível contaminação com bactérias ou outros organismos microscópicos presentes no ambiente circundante ou no material de colheita, ou com material biológico proveniente de organismos eucariotas, como sejam as células humanas em amostras colhidas para estudos do microbioma humano. Embora o genoma destes organismos não contenha sequências do gene *16S rRNA*, não é tecnicamente exequível isolar as células procariotas antes de se proceder à extracção do DNA. Desta forma, o DNA eucariota irá estar presente em menor ou maior proporção na amostra final, o que pode não só reduzir a possibilidade de detecção de espécies menos abundantes de microorganismos, mas também afectar ou enviesar a amplificação das sequências do gene *16S rRNA* por ligação inespecífica dos oligonucleótidos a sequências de origem eucariota. A forma mais eficiente de excluir as sequências eucariotas, de uma amostra de metagenómica, é através de métodos bioinformáticos que permitem alinhar as leituras em sequências de referência eucariotas e, subsequentemente, filtrar aquelas que evidenciem homologia com essas sequências, efectuando assim uma "purificação" *in silico* do DNA contaminante após sequenciação das amostras.

A amplificação dos fragmentos do gene *16S rRNA* pode também não reproduzir a abundância original dos respetivos genomas, uma vez que a selecção dos oligonucleótidos utilizados na reacção da PCR pode favorecer a amplificação de sequências de determinados grupos taxonómicos em detrimento de outros (Klindworth *et al.*, 2013). Neste sentido, é preferível amplificar mais do que uma região hipervariável do gene *16S rRNA* para garantir que os resultados não estão enviesados pela escolha do fragmento a amplificar. O enviesamento dos resultados pode também surgir em consequência da presença de bases variantes, em uma ou mais estirpes, no local de ligação de um ou ambos os oligonucleótidos. Consequentemente, a amplificação gerará um menor número de produtos para a(s) espécie(s) em causa, o que resulta numa sub-estimação da real abundância populacional. Há ainda que contar com os erros introduzidos pela enzima polimerase durante a reacção da PCR, que sobressaiem de forma mais evidente na sequenciação paralela massiva, uma vez que as leituras são analisadas uma a uma e não "em conjunto" como no caso da sequenciação tradicional. Um outro problema recorrente durante a amplificação é a formação de produtos quiméricos, ou seja, produtos de amplificação compostos por sequências originárias de 2 genomas distintos, que formam um produto único a partir da hibridação e extensão de fragmentos incompletos. Estas sequências "híbridas" podem aumentar artificialmente a diversidade microbiológica de uma determinada amostra, sugerindo a presença de

"novos" microorganismos. À semelhança do que acontece com a presença de sequências eucariotas, apenas os métodos bioinformáticos poderão identificar e filtrar eficientemente as sequências quiméricas do conjunto de dados originais.

Ao nível da sequenciação há também que contar com os erros produzidos pelos *softwares* de análise de *base-calling* das plataformas de sequenciação. Embora as leituras de baixa qualidade possam ser excluídas na fase inicial da análise de dados, eliminando assim potenciais erros de *base-calling*, as restantes leituras de boa qualidade poderão apresentar uma ou mais bases de menor qualidade que poderão constituir erros de sequenciação. Existem *softwares* que permitem detectar potenciais erros, através da aplicação de modelos de erro específicos, mas não é possível eliminar por completo erros pontuais que escapam a esses modelos. Tal como foi referido anteriormente, o pré-processamento das leituras de sequenciação é uma tarefa necessária para melhorar o alinhamento/mapeamento das leituras num genoma de referência e para a montagem de genomas. Dado o expectável aumento do número de estudos de sequenciação de metagenomas que deverá ocorrer nos próximos anos, são de esperar novas e mais completas ferramentas de avaliação de qualidade que permitam reduzir ou eliminar potenciais erros gerados pelas plataformas de sequenciação.

Ao nível da análise de dados de sequenciação do gene *16S rRNA*, existem outros factores que dificultam a classificação das leituras num determinado grupo taxonómico. O elevado número de programas de classificação taxonómica existentes no domínio público, baseados na análise de semelhança entre as leituras e as sequências de referência ou em agrupamentos de leituras que partilham entre si um determinado grau de identidade, introduz necessariamente variabilidade ao nível dos resultados. Adicionalmente, as bases de dados de sequências do gene *16S rRNA* associadas aos programas de classificação, constituem um outro factor que afecta a caracterização taxonómica dos metagenomas. Isto pode dever-se à maior ou menor completude das diferentes colecções e à qualidade da respectiva curação, quer ao nível das sequências quer ao nível da taxonomia. Os programas de análise devem também ter em conta o número de cópias do gene *16S rRNA* quando analisam amostras contendo genomas com diferente número de cópias deste gene (Stoddard *et al.*, 2015). Estes factores tornam-se ainda mais relevantes à medida que nos afastamos da raíz da árvore filogenética, uma vez que as diferenças nucleotídicas entre grupos de microorganismos tendem a esbater-se, o que coloca uma maior exigência nos métodos de classificação e dificulta a atribuição de leituras aos níveis taxonómicos inferiores. Não é assim de estranhar que muitos dos programas actualmente existentes apenas classifiquem as leituras ao nível do género ou em táxones de níveis superiores. Em síntese, não existe ainda um método de análise "gold standard" que forneça os resultados expectáveis quando se analisa um conjunto de dados de sequenciação metagenómica, quer para o gene *16S rRNA* quer para genomas completos.

No contexto da análise de leituras do gene *16S rRNA*, é de todo o interesse dispor de uma ferramenta que permita testar diferentes programas de classificação com base em sequências com taxonomia conhecida *a priori*, a fim de avaliar a performance de classificação dos vários programas. No presente trabalho, foi desenhado de raíz um programa (*sim16S*) em linguagem *MatLab*, que permite gerar conjuntos de leituras de sequenciação do gene *16S rRNA*, compostos por proporções variáveis de diferentes microorganismos com taxonomia definida. Desta forma, os resultados obtidos pelos programas de classificação taxonómica podem ser objectivamente avaliados uma vez que existe um conjunto de leituras de "referência" com o qual podem ser comparados. Além disso, o *sim16S* permite introduzir substituições de bases nas sequências do gene *16S rRNA*, em proporção definida pelo utilizador e em posições escolhidas aleatoriamente, por forma a simular a presença de erros de

sequenciação e observar a forma como as substituições afectam a performance dos programas de classificação.

Neste trabalho, o *sim16S* foi usado para produzir 17 conjuntos distintos de leituras de 2 regiões hipervariáveis do gene *16S rRNA*. Em particular, foram simulados comprimentos de leitura de 150 pb (região V3) e de 250 pb (região V4), proporções variáveis (1%, 10% e 100%) de leituras com erros e diferente número (1, 2 ou 4) de erros por leitura, para uma ou ambas as regiões hipervariáveis. Para testar a classificação taxonómica dos vários conjuntos de leituras, foram seleccionados os 2 programas de classificação mais citados na literatura científica até final de 2016. O *mothur* (Schloss *et al.*, 2009) e o *QIIME* (Caporaso *et al.*, 2010a) foram usados aplicando tutoriais específicos disponibilizados pelos autores. No entanto, ambos os programas permitem flexibilizar as abordagens de classificação e utilizar diversas parametrizações, por forma a optimizar a classificação taxonómica. Desta forma, as diferenças nos resultados obtidos pelo *mothur* e pelo *QIIME* não devem ser tomadas como definitivas, uma vez que o objectivo do trabalho não consistia em optimizar as *pipelines* de análise dos 2 programas, mas apenas observar de que forma a classificação taxonómica era afectada pela presença de leituras com erros.

A análise dos conjuntos de dados do amplicão A e do amplicão B, com ou sem erros nas leituras, revelou à partida uma diferença importante entre os dois programas. Em qualquer dos amplicões, a proporção de leituras classificadas foi superior no *QIIME* relativamente ao *mothur*. O motivo desta diferença deve-se ao valor do parâmetro que limita o número de bases ambíguas das leituras que são sujeitas a classificação pelo *mothur*, que no presente trabalho foi definido como zero (i.e., apenas foram classificadas leituras sem bases ambíguas). O objectivo foi o de evitar que aquelas leituras fossem classificadas incorrectamente, uma vez que se desconhecia à partida o número de bases ambíguas por leitura e as suas posições no amplicão. No caso do amplicão A, se fossem admitidas 2 bases incertas por leitura no máximo, o número total de leituras classificadas pelo *mothur* poderia atingir as 10000, ou seja, a totalidade de leituras produzidas pelo *sim16S*. No entanto, o número total de sequências únicas que são classificadas não parece ser muito afectado pela remoção das leituras com bases ambíguas. Por exemplo, no caso do conjunto de dados A_1k_150_1%_1_rep1, o número de leituras únicas a classificar pelo *mothur* seria de apenas 80 comparativamente às 77 leituras classificadas após remoção das leituras com bases ambíguas. Estes resultados indicam que a remoção de leituras ambíguas não teve um impacto importante na classificação taxonómica das leituras pelo *mothur*.

Ao contrário do *QIIME*, o *mothur* permite limitar a base de dados de sequências do gene *16S rRNA* a uma dada região do gene, como por exemplo a região coberta pelo amplicão A ou B. O objectivo da selecção das sequências dos amplicões é o de limitar o alinhamento das leituras a regiões pré-definidas, em vez de se utilizar a sequência completa do gene *16S rRNA*. No presente trabalho, utilizou-se o ficheiro *rep_set_aligned.fasta* produzido pelo programa *Pynast*, incluído na *pipeline* do *QIIME*, como o alinhamento múltiplo de sequências utilizado pelo *mothur* para definição da região do gene *16S rRNA*. Em alternativa, poderia ter sido usado o molde de alinhamento múltiplo incluído no *mothur* ou no *QIIME* (ficheiro '85_otus.pynast.fasta'), mas verificou-se que o alinhamento das leituras contra as sequências alinhadas destes ficheiros produzia várias regiões de alinhamento não sobreponíveis, o que impedia a determinação da região de alinhamento de cada amplicão. No entanto, uma vez que o ficheiro *rep_set_aligned.fasta* é produzido no passo final da *pipeline* do *QIIME*, com base nas sequências únicas classificadas por este programa em cada conjunto de leituras, poderia assim enviesar a classificação do *mothur* sobre um mesmo conjunto de leituras, particularmente se se tratasse de um conjunto com erros introduzidos. Por exemplo, no caso do conjunto de dados

A_1k_150_1%_1_rep1, a utilização do ficheiro *rep_set_aligned.fasta* gerado pelo *Pynast* sobre as leituras classificadas deste conjunto de dados, ou sobre as leituras do correspondente conjunto de dados sem erros de sequenciação (A_1k_150_noMutations), produziu uma mesma região de alinhamento múltiplo e o mesmo número (n=77) de sequências únicas classificadas pelo *mothur* (resultados não apresentados). A comparação foi também estendida ao conjunto de dados A_1k_150_100%_1_rep1, que contém a totalidade de leituras (n=10000) com 1 erro de sequenciação por leitura. Neste caso, a região de alinhamento múltiplo foi também igual à obtida com o conjunto de leituras sem erros de sequenciação, enquanto o número de leituras únicas classificadas variou entre 419 (ficheiro de alinhamento sem erros) e 421 (ficheiro de alinhamento com 100% de leituras com erros). No que respeita à classificação taxonómica, a utilização do ficheiro de alinhamento sem erros resultou no mesmo número total de leituras classificadas (n=9672) que o ficheiro com erros, das quais apenas 3 mereceram uma classificação distinta ao nível do género, incluindo 1 leitura adicional em *Actinobacteria_unclassified* e 1 leitura a menos nos géneros *Streptosporangiales_unclassified* e *Microbispora* (resultados não apresentados). Estes resultados mostram que a utilização de um ficheiro de alinhamento potencialmente distinto em cada conjunto de dados analisado pelo *mothur*, ao invés da utilização de um ficheiro único, não produziu diferenças relevantes na classificação taxonómica deste programa. Em resumo, a remoção das leituras com bases ambíguas e a utilização de um ficheiro de alinhamento produzido pelo *QIIME* sobre as leituras classificadas por este, não afectaram de forma significativa a classificação taxonómica do *mothur*, podendo assim ser directamente comparada com a obtida no *QIIME*. No entanto, uma vez que a remoção de leituras com bases ambíguas não é parametrizável no algoritmo utilizado no *QIIME*, a possibilidade de classificação de leituras com bases incertas pelo *mothur* poderia permitir uma comparação "mais justa" com aquele programa.

Apesar das limitações acima apresentadas, o *mothur* conseguiu aproximar-se mais do número total de táxones produzidos pelo *sim16S* do que o *QIIME*. Para qualquer dos amplicões testados, o *mothur* identificou mais táxones nos níveis taxonómicos de ordem, família e género, não existindo diferenças relativamente ao *QIIME* nos restantes níveis taxonómicos. Adicionalmente, alguns dos táxones identificados por este programa não estavam representados nos conjuntos de leituras seleccionadas pelo *sim16S* incluindo, entre outros, os géneros *Paracoccus*, *Hafnia* e *Tsukamurella*. Esta evidência poderia ser consequência da presença de sequências de amplicões iguais às daqueles géneros nas leituras do *sim16S*, mas com taxonomia distinta. Para excluir esta hipótese, foi analisado o ficheiro 'repeatSeq.txt' gerado pelo *sim16S* para o conjunto de amplicões das leituras sem erros (conjunto A_1k_150_noMutations). Este ficheiro regista o número de vezes que cada sequência de amplicão aparece no conjunto de amplicões (pré-randomização), uma vez que a mesma sequência parcial do gene *16S rRNA* pode aparecer associada a diferentes identificações taxonómicas da base de dados. As sequências dos géneros *Paracoccus* e *Hafnia* não tinham homólogos no conjunto de amplicões enquanto as do género *Tsukamurella* tinham sequências totalmente idênticas a outros amplicões, mas que estavam associadas com diferentes espécies daquele género. Desta forma conclui-se que a identificação dos géneros extra pelo *QIIME* se deve exclusivamente ao algoritmo de classificação taxonómica utilizado por este programa, o mesmo não se verificando com o *mothur*. Estes resultados mostram que o *mothur* pode oferecer um retrato mais fiel da diversidade taxonómica de um determinada comunidade do que o *QIIME*.

O conjunto de leituras sem erros do amplicão A (ficheiro 'readSet.txt' do conjunto A_1k_150_noMutations) foi utilizado como molde para a introdução de erros aleatórios com a função *mutAmp.m* do *sim16S*. Desta forma, pretendeu-se avaliar o impacto na classificação taxonómica dos conjuntos de leituras contendo aproximadamente 1%, aproximadamente 10% e 100% de leituras com 1 ou mais erros. Nos vários conjuntos de dados com erros, o *mothur* exibiu uma tendência para

aumentar o número de táxones sem classificação completa, em relação directa com a proporção de leituras com erros ou com o número de erros por leitura. Estes resultados contrastaram de forma significativa com os obtidos pelo *QIIME*, em que apenas foi afectado 1 táxon ao nível do género. O *mothur* também mostrou um maior número de leituras não classificadas ou classificadas incorrectamente, à exceção de um replicado do conjunto com 100% de leituras com erros. Neste respeito, a variação observada entre os replicados em ambos os programas deve-se ao processo aleatório de introdução de erros nas leituras, que foi realizado pelo *sim16S* em cada replicado. Este efeito foi também observado nos conjuntos independentes de dados dos 2 amplicões, contendo diferentes proporções de leituras com erros. Em ambos os amplicões, não se verificou uma propensão clara para um aumento do número de leituras sem classificação completa ou com classificação incorrecta, quando a proporção de leituras com 1 erro variou de 1% para 10%. Pelo contrário, este aumento verificou-se apenas no número de leituras sem classificação completa (para ambos os programas com o amplicão A) e com classificação incorrecta (para o *QIIME* com o amplicão B), quando os conjuntos de dados continham somente leituras com erros. Estes resultados sugerem que uma percentagem global de erros máxima de ~0.07% (amplicão A com 10% de leituras com 1 erro por leitura) não tem um impacto no número de leituras com classificação incompleta ou incorrecta. No entanto, uma percentagem superior de erros traduziu-se num aumento do número de táxones em que as proporções relativas diferiam do obtido pelo *sim16S*. Por exemplo, o número total de táxones de todos os níveis taxonómicos, que não apresentavam o número de leituras esperadas para o amplicão A, aumentou de 28 para 77 nos conjuntos de leituras sem erros e com 100% de erros de sequenciação, respectivamente. Na larga maioria destes táxones a variação foi negativa, ou seja, o número de leituras foi inferior ao esperado, tendo as leituras em défice sido atribuídas a táxones sem classificação completa e/ou a um reduzido número de táxones com classificação completa, cujas proporções relativas aumentaram assim de forma significativa. Este efeito foi particularmente visível nas leituras atribuídas por excesso pelo *QIIME* ao género *Microbispora*. Em resumo, a introdução de erros nas leituras mostrou um maior impacto ao nível das proporções relativas dos vários táxones nos conjuntos de dados, contribuindo assim para uma pior caracterização das abundâncias populacionais relativas de uma comunidade de microorganismos.

A relevância de diferentes bases de dados de sequências do gene *16S rRNA* foi também avaliada neste trabalho. Os conjuntos de dados do amplicão A com diferentes proporções de leituras com erros, foram igualmente classificados com o *mothur* e o *QIIME* usando as bases de dados de sequências e taxonomia utilizadas nos respectivos tutoriais. Neste comparativo, foi evidente o aumento da proporção de táxones sem classificação taxonómica completa, particularmente com o *mothur*. Por um lado, estes resultados eram expectáveis uma vez que a base de dados de 1000 sequências utilizada previamente para classificar os mesmos conjuntos de leituras, constituiu também o conjunto de sequências do gene *16S rRNA* a partir das quais foram geradas essas leituras. Por outro lado, seria de esperar que bases de dados mais completas, como as utilizadas por defeito no *mothur* (base de dados *SILVA* com ~15000 sequências) e no *QIIME* (base de dados *Greengenes* com ~100000 sequências), permitissem uma classificação taxonómica mais exacta dos conjuntos de dados simulados. Os resultados observados podem dever-se a diferenças na composição das sequências do gene *16S rRNA* entre bases de dados distintas ou a selecção prévia de sequências nas bases de dados, como por exemplo a versão reduzida da base de dados *SILVA* utilizada no *mothur*. Neste contexto, é interessante verificar que o *QIIME* produziu resultados significativamente melhores em termos de táxones não classificados completamente do que o *mothur*. Independentemente destas diferenças, a questão relevante é a de saber qual a base de dados a usar em determinado estudo de metagenómica e, neste aspecto, é desejável o utilizador recorrer a diferentes bases de dados para avaliar qual destas oferece os melhores resultados, o que poderá ser feito de forma simples no *mothur* uma vez que o pacote de

instalação do *software* integra bases de dados de sequências de diferentes origens.

Como já foi referido anteriormente, não constituiu um objetivo principal deste trabalho a optimização de *pipelines* de classificação taxonómica de leituras simuladas. Assim, não é legítimo tentar determinar qual dos 2 programas utilizados é o "melhor" para classificação de leituras do gene *16S rRNA* pois, como foi referido várias vezes, existe a possibilidade de alterar diferentes parâmetros da análise com o objectivo de a adequar o melhor possível aos resultados desejados. No entanto, em face dos resultados produzidos neste trabalho, efectuou-se a determinação da exactidão da classificação taxonómica dos 2 programas para os conjuntos de leituras sem erros (ambos os amplicões) e para os respectivos conjuntos do amplicão A contendo diferentes proporções de leituras com erros e diferente número de erros por leitura. O *QIIME* mostrou uma exactidão equivalente ou superior ao *mothur* na classificação das leituras sem erros de ambos os amplicões, à exceção do género no amplicão B. Este resultado deveu-se ao número de leituras sobre-estimadas em alguns táxones e à atribuição de leituras a táxones não existentes no conjunto de dados, o que não se observou no *mothur*. Nos conjuntos com 1 erro de sequenciação por leitura, a exactidão manteve-se igual ou próxima dos 100% ao nível do filo, classe e ordem nos 2 programas, independentemente da proporção de leituras com erros. No entanto, o *QIIME* conseguiu manter uma exactidão equivalente nos restantes níveis taxonómicos, independentemente do número de erros por leitura, excepto quando na presença da totalidade das leituras com erros. Em síntese, o *mothur* demonstrou globalmente uma maior sensibilidade à presença de erros nas leituras do que o *QIIME*, mas os resultados obtidos sugerem que esta vantagem pode deixar de existir se a distribuição de erros se estender a todo o conjunto de leituras e não estiver concentrada em apenas uma pequena percentagem das mesmas.

O programa *sim16S* mostrou ser uma ferramenta de grande utilidade para a criação de leituras simuladas do gene *16S rRNA*. O programa é bastante flexível, podendo usar qualquer base de dados de sequências do gene *16S rRNA*, adaptar-se à região de interesse do gene e ao comprimento das leituras geradas pelas diferentes plataformas de sequenciação (por exemplo, *Illumina*, *Ion Torrent* ou 454/*Roche*) assim como permitir modificar aleatoriamente a sequência de bases de cada leitura de forma a simular erros de sequenciação. Uma outra vantagem é a possibilidade de a primeira função do programa (*seqConc.m*) poder ser corrida uma única vez para cada base de dados utilizada, o que reduz substancialmente o tempo de execução do programa se o utilizador usar sempre a mesma base de dados de referência. No presente trabalho, o *sim16S* foi executado usando valores específicos para cada argumento do programa, como sejam a base de dados *SILVA*, o número de sequências extraídas da base de dados (1000 ou 5000), as sequências específicas de amplicões e o comprimento das leituras (150 ou 250 bases). Para o programa poder ser usado livremente com outras parametrizações, é necessário introduzir no código instruções de controlo dos parâmetros por forma a evitar erros. Por exemplo, o programa origina um erro quando na mesma entrada da base de dados *SILVA* existe mais do que uma posição possível para uma ou ambas as sequências dos oligonucleótidos, o que faz com que não seja actualmente possível extrair mais de ~7000 sequências da base de dados *SILVA*, ou quando um dos amplicões seleccionados tem um comprimento inferior ao tamanho de leitura definido. Além do gene *16S rRNA*, o programa *sim16S* pode em teoria ser usado para gerar leituras simuladas de qualquer outro gene ribossomal procariota ou eucariota, desde que exista uma correspondente base de dados de sequências de referência e sejam conhecidas as sequências dos oligonucleótidos a usar para formação dos amplicões.

Em resumo, a metagenómica é actualmente uma disciplina em grande expansão, nomeadamente nas áreas da agricultura e florestas, gestão de recursos hídricos, produção animal e medicina humana, que beneficiou enormemente dos desenvolvimentos tecnológicos ocorridos nos últimos anos ao nível

das plataformas de sequenciação. Esta expansão tem tendência para aumentar ainda mais nos próximos anos, uma vez que as plataformas de sequenciação tenderão progressivamente a produzir maiores volumes de dados a custos cada vez mais reduzidos. Ao permitirem conhecer a composição das comunidades de microorganismos existentes em qualquer nicho ambiental, os estudos de metagenómica darão um contributo inestimável para um melhor conhecimento das interacções entre os organismos vivos e entre estes e o meio ambiente que os rodeia.

6. Referências bibliográficas

Abubucker, S. et al. (2012) ‘Metabolic reconstruction for metagenomic data and its application to the human microbiome.’, PLoS computational biology. United States, 8(6), p. e1002358. doi: 10.1371/journal.pcbi.1002358.

Afiahayati, Sato, K. and Sakakibara, Y. (2015) ‘MetaVelvet-SL: an extension of the Velvet assembler to a de novo metagenomic assembler utilizing supervised learning.’, DNA research : an international journal for rapid publication of reports on genes and genomes. England, 22(1), pp. 69–77. doi: 10.1093/dnares/dsu041.

Ahn, T.-H., Chai, J. and Pan, C. (2015) ‘Sigma: strain-level inference of genomes from metagenomic analysis for biosurveillance.’, Bioinformatics (Oxford, England). England, 31(2), pp. 170–177. doi: 10.1093/bioinformatics/btu641.

Albanese, D. et al. (2015) ‘MICCA: a complete and accurate software for taxonomic profiling of metagenomic data.’, Scientific reports. England, 5, p. 9743. doi: 10.1038/srep09743.

Alneberg, J. et al. (2014) ‘Binning metagenomic contigs by coverage and composition.’, Nature methods. United States, 11(11), pp. 1144–1146. doi: 10.1038/nmeth.3103.

Alonso-Alemany, D. et al. (2014) ‘Further steps in TANGO: improved taxonomic assignment in metagenomics.’, Bioinformatics (Oxford, England). England, 30(1), pp. 17–23. doi: 10.1093/bioinformatics/btt256.

Altschul, S. F. et al. (1990) ‘Basic local alignment search tool.’, Journal of molecular biology. England, 215(3), pp. 403–410. doi: 10.1016/S0022-2836(05)80360-2.

Ames, S. K. et al. (2013) ‘Scalable metagenomic taxonomy classification using a reference genome database.’, Bioinformatics (Oxford, England). England, 29(18), pp. 2253–2260. doi: 10.1093/bioinformatics/btt389.

Anders, S., Pyl, P. T. and Huber, W. (2015) ‘HTSeq--a Python framework to work with high-throughput sequencing data.’, Bioinformatics (Oxford, England). England, 31(2), pp. 166–169. doi: 10.1093/bioinformatics/btu638.

Andrews, S. (2010) FastQC: A quality control tool for high throughput sequence data. Available at: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/ (Accessed: 2 August 2017).

Asshauer, K. P. et al. (2015) ‘Tax4Fun: predicting functional profiles from metagenomic 16S rRNA data.’, Bioinformatics (Oxford, England). England, 31(17), pp. 2882–2884. doi: 10.1093/bioinformatics/btv287.

Backhed, F. et al. (2005) ‘Host-bacterial mutualism in the human intestine.’, Science (New York, N.Y.). United States, 307(5717), pp. 1915–1920. doi: 10.1126/science.1104816.

Bankevich, A. et al. (2012) ‘SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing.’, Journal of computational biology : a journal of computational molecular cell biology. United States, 19(5), pp. 455–477. doi: 10.1089/cmb.2012.0021.

Bengtsson, J. et al. (2012) ‘Megraft: a software package to graft ribosomal small subunit (16S/18S) fragments onto full-length sequences for accurate species richness and sequencing depth analysis in pyrosequencing-length metagenomes and similar environmental datasets.’, Research in microbiology. France, 163(6–7), pp. 407–412. doi: 10.1016/j.resmic.2012.07.001.

Bengtsson-Palme, J. et al. (2015) ‘METAXA2: improved identification and taxonomic classification of small and large subunit rRNA in metagenomic data.’, Molecular ecology resources. England, 15(6), pp. 1403–1414. doi: 10.1111/1755-0998.12399.

Bentley, S. D. and Parkhill, J. (2004) ‘Comparative genomic structure of prokaryotes.’, Annual review of genetics. United States, 38, pp. 771–792. doi: 10.1146/annurev.genet.38.072902.094318.

Bhaduri, A. et al. (2012) ‘Rapid identification of non-human sequences in high-throughput sequencing datasets.’, Bioinformatics (Oxford, England). England, 28(8), pp. 1174–1175. doi: 10.1093/bioinformatics/bts100.

Boisvert, S. et al. (2012) ‘Ray Meta: scalable de novo metagenome assembly and profiling.’, Genome biology. England, 13(12), p. R122. doi: 10.1186/gb-2012-13-12-r122.

Boisvert, S., Laviolette, F. and Corbeil, J. (2010) ‘Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies.’, Journal of computational biology : a journal of computational molecular cell biology. United States, 17(11), pp. 1519–1533. doi: 10.1089/cmb.2009.0238.

Bokulich, N. A. et al. (2013) ‘Quality-filtering vastly improves diversity estimates from Illumina amplicon sequencing.’, Nature methods. United States, 10(1), pp. 57–59. doi: 10.1038/nmeth.2276.

Bolger, A. M., Lohse, M. and Usadel, B. (2014) ‘Trimmomatic: a flexible trimmer for Illumina sequence data.’, Bioinformatics (Oxford, England). England, 30(15), pp. 2114–2120. doi: 10.1093/bioinformatics/btu170.

Bose, T. et al. (2015) ‘COGNIZER: A Framework for Functional Annotation of Metagenomic Datasets.’, PloS one. United States, 10(11), p. e0142102. doi: 10.1371/journal.pone.0142102.

Bowe, A., Onodera, T., Sadakane, K., Shibuya, T. (2012) ‘Succinct de Bruijn Graphs’, in Raphael, B., Tang, J. (ed.) Algorithms in Bioinformatics. WABI 2012. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 225–235. doi: https://doi.org/10.1007/978-3-642-33122-0_18.

Bradley, R. D. and Hillis, D. M. (1997) ‘Recombinant DNA sequences generated by PCR amplification.’, Molecular biology and evolution. United States, pp. 592–593.

Brady, A. and Salzberg, S. (2011) ‘PhymmBL expanded: confidence scores, custom databases, parallelization and more.’, Nature methods. United States, p. 367. doi: 10.1038/nmeth0511-367.

Brady, A. and Salzberg, S. L. (2009) ‘Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models.’, Nature methods. United States, 6(9), pp. 673–676. doi: 10.1038/nmeth.1358.

Brinda, K., Sykulski, M. and Kucherov, G. (2015) ‘Spaced seeds improve k-mer-based metagenomic classification.’, Bioinformatics (Oxford, England). England, 31(22), pp. 3584–3592. doi: 10.1093/bioinformatics/btv419.

Cai, Y. and Sun, Y. (2011) ‘ESPRIT-Tree: hierarchical clustering analysis of millions of 16S rRNA pyrosequences in quasilinear computational time.’, Nucleic acids research. England, 39(14), p. e95. doi: 10.1093/nar/gkr349.

Campbell, A., Mrazek, J. and Karlin, S. (1999) ‘Genome signature comparisons among prokaryote, plasmid, and mitochondrial DNA.’, Proceedings of the National Academy of Sciences of the United States of America. United States, 96(16), pp. 9184–9189.

Caporaso, J. G. et al. (2010a) ‘QIIME allows analysis of high-throughput community sequencing data.’, Nature methods. United States, pp. 335–336. doi: 10.1038/nmeth.f.303.

Caporaso, J. G. et al. (2010b) ‘PyNAST: a flexible tool for aligning sequences to a template alignment.’, Bioinformatics (Oxford, England). England, 26(2), pp. 266–267. doi: 10.1093/bioinformatics/btp636.

Chatterjee, S. et al. (2014) ‘SEK: sparsity exploiting k-mer-based estimation of bacterial community composition.’, Bioinformatics (Oxford, England). England, 30(17), pp. 2423–2431. doi: 10.1093/bioinformatics/btu320.

Chaudhary, N. et al. (2015) ‘16S classifier: a tool for fast and accurate taxonomic classification of 16S rRNA hypervariable regions in metagenomic datasets.’, PloS one. United States, 10(2), p. e0116106. doi: 10.1371/journal.pone.0116106.

Chen, C. et al. (2014) ‘Software for pre-processing Illumina next-generation sequencing short read sequences.’, Source code for biology and medicine. England, 9, p. 8. doi: 10.1186/1751-0473-9-8.

Chen, T. et al. (2010) ‘The Human Oral Microbiome Database: a web accessible resource for investigating oral microbe taxonomic and genomic information.’, Database : the journal of biological databases and curation. England, 2010, p. baq013. doi: 10.1093/database/baq013.

Chen, W. et al. (2013) ‘MSClust: A Multi-Seeds based Clustering algorithm for microbiome profiling using 16S rRNA sequence.’, Journal of microbiological methods. Netherlands, 94(3), pp. 347–355. doi: 10.1016/j.mimet.2013.07.004.

Chun, J. et al. (2007) ‘EzTaxon: a web-based tool for the identification of prokaryotes based on 16S ribosomal RNA gene sequences.’, International journal of systematic and evolutionary microbiology. England, 57(Pt 10), pp. 2259–2261. doi: 10.1099/ijs.0.64915-0.

Cleary, B. et al. (2015) ‘Detection of low-abundance bacterial strains in metagenomic datasets by eigengenome partitioning.’, Nature biotechnology. United States, 33(10), pp. 1053–1060. doi: 10.1038/nbt.3329.

Cole, J. R. et al. (2003) ‘The Ribosomal Database Project (RDP-II): previewing a new autoaligner that allows regular updates and the new prokaryotic taxonomy.’, Nucleic acids research. England, 31(1), pp. 442–443.

Cole, J. R. et al. (2005) ‘The Ribosomal Database Project (RDP-II): sequences and tools for high-throughput rRNA analysis.’, Nucleic acids research. England, 33(Database issue), pp. D294-6. doi: 10.1093/nar/gki038.

Cole, J. R. et al. (2007) ‘The ribosomal database project (RDP-II): introducing myRDP space and quality controlled public data.’, Nucleic acids research. England, 35(Database issue), pp. D169-72. doi: 10.1093/nar/gkl889.

Cole, J. R. et al. (2009) ‘The Ribosomal Database Project: improved alignments and new tools for rRNA analysis.’, Nucleic acids research. England, 37(Database issue), pp. D141-5. doi: 10.1093/nar/gkn879.

Cole, J. R. et al. (2014) ‘Ribosomal Database Project: data and tools for high throughput rRNA analysis.’, Nucleic acids research. England, 42(Database issue), pp. D633-42. doi: 10.1093/nar/gkt1244.

Criscuolo, A. and Brisse, S. (2013) ‘AlienTrimmer: a tool to quickly and accurately trim off multiple short contaminant sequences from high-throughput sequencing reads.’, Genomics. United States, 102(5–6), pp. 500–506. doi: 10.1016/j.ygeno.2013.07.011.

Darling, A. E. et al. (2014) ‘PhyloSift: phylogenetic analysis of genomes and metagenomes.’, PeerJ. United States, 2, p. e243. doi: 10.7717/peerj.243.

Davenport, C. F. et al. (2012) ‘Genometa--a fast and accurate classifier for short metagenomic shotgun reads.’, PloS one. United States, 7(8), p. e41224. doi: 10.1371/journal.pone.0041224.

Davis, M. P. A. et al. (2013) ‘Kraken: a set of tools for quality control and analysis of high-throughput sequence data.’, Methods (San Diego, Calif.). United States, 63(1), pp. 41–49. doi: 10.1016/j.ymeth.2013.06.027.

De Rijk, P. et al. (2000) ‘The European large subunit ribosomal RNA database.’, Nucleic acids research. England, 28(1), pp. 177–178.

DeSantis, T. Z. et al. (2006a) ‘Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB.’, Applied and environmental microbiology. United States, 72(7), pp. 5069–5072. doi: 10.1128/AEM.03006-05.

DeSantis, T. Z. J. et al. (2006b) ‘NAST: a multiple sequence alignment server for comparative analysis of 16S rRNA genes.’, Nucleic acids research. England, 34(Web Server issue), pp. W394-9. doi: 10.1093/nar/gkl244.

Dewhirst, F. E. et al. (2010) ‘The human oral microbiome.’, Journal of bacteriology. United States, 192(19), pp. 5002–5017. doi: 10.1128/JB.00542-10.

Diaz, N. N. et al. (2009) ‘TACOA: taxonomic classification of environmental genomic fragments

using a kernelized nearest neighbor approach.', BMC bioinformatics. England, 10, p. 56. doi: 10.1186/1471-2105-10-56.

Dimon, M. T. et al. (2013) 'IMSA: integrated metagenomic sequence analysis for identification of exogenous reads in a host genomic background.', PloS one. United States, 8(5), p. e64546. doi: 10.1371/journal.pone.0064546.

Ding, C. and Peng, H. (2005) 'Minimum redundancy feature selection from microarray gene expression data.', Journal of bioinformatics and computational biology. England, 3(2), pp. 185–205.

Dodt, M. et al. (2012) 'FLEXBAR-Flexible Barcode and Adapter Processing for Next-Generation Sequencing Platforms.', Biology. Switzerland, 1(3), pp. 895–905. doi: 10.3390/biology1030895.

Droge, J., Gregor, I. and McHardy, A. C. (2015) 'Taxator-tk: precise taxonomic assignment of metagenomes by fast approximation of evolutionary neighborhoods.', Bioinformatics (Oxford, England). England, 31(6), pp. 817–824. doi: 10.1093/bioinformatics/btu745.

Eddy, S. R. (2011) 'Accelerated Profile HMM Searches.', PLoS computational biology. United States, 7(10), p. e1002195. doi: 10.1371/journal.pcbi.1002195.

Edgar, R. C. (2004) 'MUSCLE: a multiple sequence alignment method with reduced time and space complexity.', BMC bioinformatics. England, 5, p. 113. doi: 10.1186/1471-2105-5-113.

Edgar, R. C. (2010) 'Search and clustering orders of magnitude faster than BLAST.', Bioinformatics (Oxford, England). England, 26(19), pp. 2460–2461. doi: 10.1093/bioinformatics/btq461.

Edgar, R. C. et al. (2011) 'UCHIME improves sensitivity and speed of chimera detection.', Bioinformatics (Oxford, England). England, 27(16), pp. 2194–2200. doi: 10.1093/bioinformatics/btr381.

Erdmann, V. A. et al. (1983) 'Collection of published 5S and 5.8S ribosomal RNA sequences.', Nucleic acids research. England, 11(1), pp. r105-33.

European Ribossomal RNA Database (2002). Available at: http://bioinformatics.psb.ugent.be/webtools/rRNA/software/index.html (Accessed: 29 July 2017).

Ewing, B. and Green, P. (1998) 'Base-calling of automated sequencer traces using phred. II. Error probabilities.', Genome research. United States, 8(3), pp. 186–194.

Ewing, B. et al. (1998) 'Base-calling of automated sequencer traces using phred. I. Accuracy assessment.', Genome research. United States, 8(3), pp. 175–185.

FASTX-Toolkit: FASTQ/A short-reads pre-processing tools (2009). Available at: http://hannonlab.cshl.edu/fastx_toolkit/ (Accessed: 2 August 2017).

Finn, R. D. et al. (2010) 'The Pfam protein families database.', Nucleic acids research. England, 38(Database issue), pp. D211-22. doi: 10.1093/nar/gkp985.

Fofanov, Y. et al. (2004) 'How independent are the appearances of n-mers in different genomes?', Bioinformatics (Oxford, England). England, 20(15), pp. 2421–2428. doi: 10.1093/bioinformatics/bth266.

Fox, G. E. et al. (1977) 'Classification of methanogenic bacteria by 16S ribosomal RNA characterization.', Proceedings of the National Academy of Sciences of the United States of America. United States, 74(10), pp. 4537–4541.

Ghosh, T. S. et al. (2012) 'C16S - a Hidden Markov Model based algorithm for taxonomic classification of 16S rRNA gene sequences.', Genomics. United States, 99(4), pp. 195–201. doi: 10.1016/j.ygeno.2012.01.008.

Ghosh, T. S., Monzoorul Haque, M. and Mande, S. S. (2010) 'DiScRIBinATE: a rapid method for accurate taxonomic classification of metagenomic sequences.', BMC bioinformatics. England, 11 Suppl 7, p. S14. doi: 10.1186/1471-2105-11-S7-S14.

Gilbert, J. A. et al. (2008) 'Detection of large numbers of novel sequences in the metatranscriptomes of complex marine microbial communities.', PloS one. United States, 3(8), p. e3042. doi:

10.1371/journal.pone.0003042.
Greuter, D. et al. (2016) ‘probeBase--an online resource for rRNA-targeted oligonucleotide probes and primers: new features 2016.’, Nucleic acids research. England, 44(D1), pp. D586-9. doi: 10.1093/nar/gkv1232.
Griffen, A. L. et al. (2011) ‘CORE: a phylogenetically-curated 16S rDNA database of the core oral microbiome.’, PloS one. United States, 6(4), p. e19051. doi: 10.1371/journal.pone.0019051.
Guo, X. et al. (2015) ‘DIME: a novel framework for de novo metagenomic sequence assembly.’, Journal of computational biology : a journal of computational molecular cell biology. United States, 22(2), pp. 159–177. doi: 10.1089/cmb.2014.0251.
Haft, D. H., Selengut, J. D. and White, O. (2003) ‘The TIGRFAMs database of protein families.’, Nucleic acids research. England, 31(1), pp. 371–373.
Haider, B. et al. (2014) ‘Omega: an overlap-graph de novo assembler for metagenomics.’, Bioinformatics (Oxford, England). England, 30(19), pp. 2717–2722. doi: 10.1093/bioinformatics/btu395.
Handelsman, J. et al. (1998) ‘Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products.’, Chemistry & biology. United States, 5(10), pp. R245-9.
Hao, X., Jiang, R. and Chen, T. (2011) ‘Clustering 16S rRNA for OTU prediction: a method of unsupervised Bayesian clustering.’, Bioinformatics (Oxford, England). England, 27(5), pp. 611–618. doi: 10.1093/bioinformatics/btq725.
Henry, V. J. et al. (2014) ‘OMICtools: an informative directory for multi-omic data analysis.’, Database : the journal of biological databases and curation. England, 2014. doi: 10.1093/database/bau069.
Holliday, G. L. et al. (2017) ‘Evaluating Functional Annotations of Enzymes Using the Gene Ontology.’, Methods in molecular biology (Clifton, N.J.). United States, 1446, pp. 111–132. doi: 10.1007/978-1-4939-3743-1_9.
Hong, C. et al. (2014b) ‘PathoScope 2.0: a complete computational framework for strain identification in environmental or clinical sequencing samples.’, Microbiome. England, 2, p. 33. doi: 10.1186/2049-2618-2-33.
Hong, C., Manimaran, S. and Johnson, W. E. (2014a) ‘PathoQC: Computationally Efficient Read Preprocessing and Quality Control for High-Throughput Sequencing Data Sets.’, Cancer informatics. United States, 13(Suppl 1), pp. 167–176. doi: 10.4137/CIN.S13890.
Huang, Y., Gilna, P. and Li, W. (2009) ‘Identification of ribosomal RNA genes in metagenomic fragments.’, Bioinformatics (Oxford, England). England, 25(10), pp. 1338–1340. doi: 10.1093/bioinformatics/btp161.
Hugenholtz, P., Goebel, B. M. and Pace, N. R. (1998) ‘Impact of culture-independent studies on the emerging phylogenetic view of bacterial diversity.’, Journal of bacteriology. United States, 180(18), pp. 4765–4774.
Huson, D. H. et al. (2007) ‘MEGAN analysis of metagenomic data.’, Genome research. United States, 17(3), pp. 377–386. doi: 10.1101/gr.5969107.
Huson, D. H. et al. (2011) ‘Integrative analysis of environmental sequences using MEGAN4.’, Genome research. United States, 21(9), pp. 1552–1560. doi: 10.1101/gr.120618.111.
Hwang, K. et al. (2013) ‘CLUSTOM: a novel method for clustering 16S rRNA next generation sequences by overlap minimization.’, PloS one. United States, 8(5), p. e62623. doi: 10.1371/journal.pone.0062623.
Illumina Overview Tutorial: Moving Pictures of the Human Microbiome (2015). Available at: http://nbviewer.jupyter.org/github/biocore/qiime/blob/1.9.1/examples/ipynb/illumina_overview_tutorial.ipynb (Accessed: 5 June 2017).

Imelfort, M. et al. (2014) ‘GroopM: an automated tool for the recovery of population genomes from related metagenomes.’, PeerJ. United States, 2, p. e603. doi: 10.7717/peerj.603.

Jia, P. et al. (2011) ‘MetaBinG: using GPUs to accelerate metagenomic sequence classification.’, PloS one. United States, 6(11), p. e25353. doi: 10.1371/journal.pone.0025353.

Jiang, H. et al. (2012) ‘A statistical framework for accurate taxonomic assignment of metagenomic sequencing reads.’, PloS one. United States, 7(10), p. e46450. doi: 10.1371/journal.pone.0046450.

Jiang, H. et al. (2014) ‘Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads.’, BMC bioinformatics. England, 15, p. 182. doi: 10.1186/1471-2105-15-182.

Kaminski, J. et al. (2015) ‘High-Specificity Targeted Functional Profiling in Microbial Communities with ShortBRED.’, PLoS computational biology. United States, 11(12), p. e1004557. doi: 10.1371/journal.pcbi.1004557.

Kanehisa, M. and Goto, S. (2000) ‘KEGG: kyoto encyclopedia of genes and genomes.’, Nucleic acids research. England, 28(1), pp. 27–30.

Kanehisa, M. et al. (2008) ‘KEGG for linking genomes to life and the environment.’, Nucleic acids research. England, 36(Database issue), pp. D480-4. doi: 10.1093/nar/gkm882.

Kang, D. D. et al. (2015) ‘MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities.’, PeerJ. United States, 3, p. e1165. doi: 10.7717/peerj.1165.

Karlin, S. (2001) ‘Detecting anomalous gene clusters and pathogenicity islands in diverse bacterial genomes.’, Trends in microbiology. England, 9(7), pp. 335–343.

Karlin, S., Mrazek, J. and Campbell, A. M. (1997) ‘Compositional biases of bacterial genomes and evolutionary implications.’, Journal of bacteriology. United States, 179(12), pp. 3899–3913.

Katoh, K. et al. (2005) ‘MAFFT version 5: improvement in accuracy of multiple sequence alignment.’, Nucleic acids research. England, 33(2), pp. 511–518. doi: 10.1093/nar/gki198.

Kelley, D. R. et al. (2012) ‘Gene prediction with Glimmer for metagenomic sequences augmented by classification and clustering.’, Nucleic acids research. England, 40(1), p. e9. doi: 10.1093/nar/gkr1067.

Kent, W. J. (2002) ‘BLAT--the BLAST-like alignment tool.’, Genome research. United States, 12(4), pp. 656–664.

Kielbasa, S. M. et al. (2011) ‘Adaptive seeds tame genomic sequence comparison.’, Genome research. United States, 21(3), pp. 487–493. doi: 10.1101/gr.113985.110.

Kim, O.-S. et al. (2012) ‘Introducing EzTaxon-e: a prokaryotic 16S rRNA gene sequence database with phylotypes that represent uncultured species.’, International journal of systematic and evolutionary microbiology. England, 62(Pt 3), pp. 716–721. doi: 10.1099/ijs.0.038075-0.

Klappenbach, J. A. et al. (2001) ‘rrndb: the Ribosomal RNA Operon Copy Number Database.’, Nucleic acids research. England, 29(1), pp. 181–184.

Klindworth, A. et al. (2013) ‘Evaluation of general 16S ribosomal RNA gene PCR primers for classical and next-generation sequencing-based diversity studies.’, Nucleic acids research. England, 41(1), p. e1. doi: 10.1093/nar/gks808.

Kong, Y. (2011) ‘Btrim: a fast, lightweight adapter and quality trimming program for next-generation sequencing technologies.’, Genomics. United States, 98(2), pp. 152–153. doi: 10.1016/j.ygeno.2011.05.009.

Koslicki, D. et al. (2015) ‘ARK: Aggregation of Reads by K-Means for Estimation of Bacterial Community Composition.’, PloS one. United States, 10(10), p. e0140644. doi: 10.1371/journal.pone.0140644.

Koslicki, D., Foucart, S. and Rosen, G. (2013) ‘Quikr: a method for rapid reconstruction of bacterial communities via compressive sensing.’, Bioinformatics (Oxford, England). England, 29(17), pp. 2096–2102. doi: 10.1093/bioinformatics/btt336.

Koslicki, D., Foucart, S. and Rosen, G. (2014) 'WGSQuikr: fast whole-genome shotgun metagenomic classification.', PloS one. United States, 9(3), p. e91784. doi: 10.1371/journal.pone.0091784.

Kostic, A. D. et al. (2011) 'PathSeq: software to identify or discover microbes by deep sequencing of human tissue.', Nature biotechnology. United States, pp. 393–396. doi: 10.1038/nbt.1868.

Kozich, J. J. et al. (2013) 'Development of a dual-index sequencing strategy and curation pipeline for analyzing amplicon sequence data on the MiSeq Illumina sequencing platform.', Applied and environmental microbiology. United States, 79(17), pp. 5112–5120. doi: 10.1128/AEM.01043-13.

Krueger, F. (2012) Trim Galore! Available at: http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/ (Accessed: 2 August 2017).

Kultima, J. R. et al. (2016) 'MOCAT2: a metagenomic assembly, annotation and profiling framework.', Bioinformatics (Oxford, England). England, 32(16), pp. 2520–2523. doi: 10.1093/bioinformatics/btw183.

Lagesen, K. et al. (2007) 'RNAmmer: consistent and rapid annotation of ribosomal RNA genes.', Nucleic acids research. England, 35(9), pp. 3100–3108. doi: 10.1093/nar/gkm160.

Lai, B. et al. (2015) 'InteMAP: Integrated metagenomic assembly pipeline for NGS short reads.', BMC bioinformatics. England, 16, p. 244. doi: 10.1186/s12859-015-0686-x.

Langille, M. G. I. et al. (2013) 'Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences.', Nature biotechnology. United States, 31(9), pp. 814–821. doi: 10.1038/nbt.2676.

Langmead, B. and Salzberg, S. L. (2012) 'Fast gapped-read alignment with Bowtie 2.', Nature methods. United States, 9(4), pp. 357–359. doi: 10.1038/nmeth.1923.

Langmead, B. et al. (2009) 'Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.', Genome biology. England, 10(3), p. R25. doi: 10.1186/gb-2009-10-3-r25.

Larsen, N. et al. (1993) 'The ribosomal database project.', Nucleic acids research. England, 21(13), pp. 3021–3023.

Laserson, J., Jojic, V. and Koller, D. (2011) 'Genovo: de novo assembly for metagenomes.', Journal of computational biology : a journal of computational molecular cell biology. United States, 18(3), pp. 429–443. doi: 10.1089/cmb.2010.0244.

Lassmann, T., Hayashizaki, Y. and Daub, C. O. (2011) 'SAMStat: monitoring biases in next generation sequencing data.', Bioinformatics (Oxford, England). England, 27(1), pp. 130–131. doi: 10.1093/bioinformatics/btq614.

Le, V. Van, Tran, L. Van and Tran, H. Van (2016) 'A novel semi-supervised algorithm for the taxonomic assignment of metagenomic reads.', BMC bioinformatics. England, 17, p. 22. doi: 10.1186/s12859-015-0872-x.

Lee, A. Y., Lee, C. S. and Van Gelder, R. N. (2016) 'Scalable metagenomics alignment research tool (SMART): a scalable, rapid, and complete search heuristic for the classification of metagenomic sequences from complex sequence populations.', BMC bioinformatics. England, 17, p. 292. doi: 10.1186/s12859-016-1159-6.

Lee, C., Grasso, C. and Sharlow, M. F. (2002) 'Multiple sequence alignment using partial order graphs.', Bioinformatics (Oxford, England). England, 18(3), pp. 452–464.

Lee, Z. M.-P., Bussema, C. 3rd and Schmidt, T. M. (2009) 'rrnDB: documenting the number of rRNA and tRNA genes in bacteria and archaea.', Nucleic acids research. England, 37(Database issue), pp. D489-93. doi: 10.1093/nar/gkn689.

Li, D. et al. (2016) 'MEGAHIT v1.0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices.', Methods (San Diego, Calif.). United States, 102, pp. 3–11. doi: 10.1016/j.ymeth.2016.02.020.

Li, H. and Durbin, R. (2009) ‘Fast and accurate short read alignment with Burrows-Wheeler transform.’, Bioinformatics (Oxford, England). England, 25(14), pp. 1754–1760. doi: 10.1093/bioinformatics/btp324.

Li, H. et al. (2009) ‘The Sequence Alignment/Map format and SAMtools.’, Bioinformatics (Oxford, England). England, 25(16), pp. 2078–2079. doi: 10.1093/bioinformatics/btp352.

Li, R. et al. (2009) ‘SOAP2: an improved ultrafast tool for short read alignment.’, Bioinformatics (Oxford, England). England, 25(15), pp. 1966–1967. doi: 10.1093/bioinformatics/btp336.

Li, Y.-L. et al. (2015) ‘PEAT: an intelligent and efficient paired-end sequencing adapter trimming algorithm.’, BMC bioinformatics. England, 16 Suppl 1, p. S2. doi: 10.1186/1471-2105-16-S1-S2.

Liao, R. et al. (2014) ‘A New Unsupervised Binning Approach for Metagenomic Sequences Based on N-grams and Automatic Feature Weighting.’, IEEE/ACM transactions on computational biology and bioinformatics. United States, 11(1), pp. 42–54. doi: 10.1109/TCBB.2013.137.

Lin, H.-H. and Liao, Y.-C. (2016) ‘Accurate binning of metagenomic contigs via automated clustering sequences using information of genomic signatures and marker genes.’, Scientific reports. England, 6, p. 24175. doi: 10.1038/srep24175.

Lindner, M. S. and Renard, B. Y. (2013) ‘Metagenomic abundance estimation and diagnostic testing on species level.’, Nucleic acids research. England, 41(1), p. e10. doi: 10.1093/nar/gks803.

Liu, B. et al. (2011) ‘Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences.’, BMC genomics. England, 12 Suppl 2, p. S4. doi: 10.1186/1471-2164-12-S2-S4.

Lozupone, C. and Knight, R. (2005) ‘UniFrac: a new phylogenetic method for comparing microbial communities.’, Applied and environmental microbiology. United States, 71(12), pp. 8228–8235. doi: 10.1128/AEM.71.12.8228-8235.2005.

Luo, C. et al. (2015) ‘ConStrains identifies microbial strains in metagenomic datasets.’, Nature biotechnology. United States, 33(10), pp. 1045–1052. doi: 10.1038/nbt.3319.

Ma, X. et al. (2012) ‘A highly efficient and effective motif discovery method for ChIP-seq/ChIP-chip data using positional information.’, Nucleic acids research. England, 40(7), p. e50. doi: 10.1093/nar/gkr1135.

MacDonald, N. J., Parks, D. H. and Beiko, R. G. (2012) ‘Rapid identification of high-confidence taxonomic assignments for metagenomic data.’, Nucleic acids research. England, 40(14), p. e111. doi: 10.1093/nar/gks335.

Mahe, F. et al. (2015) ‘Swarm v2: highly-scalable and high-resolution amplicon clustering.’, PeerJ. United States, 3, p. e1420. doi: 10.7717/peerj.1420.

Maidak, B. L. et al. (1994) ‘The Ribosomal Database Project.’, Nucleic acids research. England, 22(17), pp. 3485–3487.

Maidak, B. L. et al. (1996) ‘The Ribosomal Database Project (RDP).’, Nucleic acids research. England, 24(1), pp. 82–85.

Maidak, B. L. et al. (1997) ‘The RDP (Ribosomal Database Project).’, Nucleic acids research. England, 25(1), pp. 109–111.

Maidak, B. L. et al. (1999) ‘A new version of the RDP (Ribosomal Database Project).’, Nucleic acids research. England, 27(1), pp. 171–173.

Maidak, B. L. et al. (2000) ‘The RDP (Ribosomal Database Project) continues.’, Nucleic acids research. England, 28(1), pp. 173–174.

Maidak, B. L. et al. (2001) ‘The RDP-II (Ribosomal Database Project).’, Nucleic acids research. England, 29(1), pp. 173–174.

Markowitz, V. M. et al. (2010) ‘The integrated microbial genomes system: an expanding comparative analysis resource.’, Nucleic acids research. England, 38(Database issue), pp. D382-90. doi: 10.1093/nar/gkp887.

Markowitz, V. M. et al. (2012) ‘IMG: the Integrated Microbial Genomes database and comparative

analysis system.', Nucleic acids research. England, 40(Database issue), pp. D115-22. doi: 10.1093/nar/gkr1044.

Martin, M. (2011) 'Cutadapt removes adapter sequences from high-throughput sequencing reads', EMBnet.journal, 17(1), pp. 10–12. doi: http://dx.doi.org/10.14806/ej.17.1.200.

Matsen, F. A., Kodner, R. B. and Armbrust, E. V. (2010) 'pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree.', BMC bioinformatics. England, 11, p. 538. doi: 10.1186/1471-2105-11-538.

McDonald, D. et al. (2012) 'An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea.', The ISME journal. England, 6(3), pp. 610–618. doi: 10.1038/ismej.2011.139.

Meinicke, P., Asshauer, K. P. and Lingner, T. (2011) 'Mixture models for analysis of the taxonomic composition of metagenomes.', Bioinformatics (Oxford, England). England, 27(12), pp. 1618–1624. doi: 10.1093/bioinformatics/btr266.

Menzel, P., Ng, K. L. and Krogh, A. (2016) 'Fast and sensitive taxonomic classification for metagenomics with Kaiju.', Nature communications. England, 7, p. 11257. doi: 10.1038/ncomms11257.

Miller, J. R. et al. (2008) 'Aggressive assembly of pyrosequencing reads with mates.', Bioinformatics (Oxford, England). England, 24(24), pp. 2818–2824. doi: 10.1093/bioinformatics/btn548.

MiSeq SOP (2013). Available at: https://mothur.org/wiki/MiSeq_SOP (Accessed: 25 September 2017).

Mohammed, M. H. et al. (2011) 'INDUS - a composition-based approach for rapid and accurate taxonomic classification of metagenomic sequences.', BMC genomics. England, 12 Suppl 3, p. S4. doi: 10.1186/1471-2164-12-S3-S4.

Mohammed, M. H. et al. (2011) 'i-rDNA: alignment-free algorithm for rapid in silico detection of ribosomal gene fragments from metagenomic sequence data sets.', BMC genomics. England, 12 Suppl 3, p. S12. doi: 10.1186/1471-2164-12-S3-S12.

Monzoorul Haque, M. et al. (2009) 'SOrt-ITEMS: Sequence orthology based approach for improved taxonomic estimation of metagenomic sequences.', Bioinformatics (Oxford, England). England, 25(14), pp. 1722–1730. doi: 10.1093/bioinformatics/btp317.

Mysara, M. et al. (2015) 'CATCh, an ensemble classifier for chimera detection in 16S rRNA sequencing studies.', Applied and environmental microbiology. United States, 81(5), pp. 1573–1584. doi: 10.1128/AEM.02896-14.

Mysara, M. et al. (2016) 'IPED: a highly efficient denoising tool for Illumina MiSeq Paired-end 16S rRNA gene amplicon sequencing data.', BMC bioinformatics. England, 17(1), p. 192. doi: 10.1186/s12859-016-1061-2.

Naccache, S. N. et al. (2014) 'A cloud-compatible bioinformatics pipeline for ultrarapid pathogen identification from next-generation sequencing of clinical samples.', Genome research. United States, 24(7), pp. 1180–1192. doi: 10.1101/gr.171934.113.

Naeem, R., Rashid, M. and Pain, A. (2013) 'READSCAN: a fast and scalable pathogen discovery program with accurate genome relative abundance estimation.', Bioinformatics (Oxford, England). England, 29(3), pp. 391–392. doi: 10.1093/bioinformatics/bts684.

Nagpal, S., Haque, M. M. and Mande, S. S. (2016) 'Vikodak--A Modular Framework for Inferring Functional Potential of Microbial Communities from 16S Metagenomic Datasets.', PloS one. United States, 11(2), p. e0148347. doi: 10.1371/journal.pone.0148347.

Nalbantoglu, O. U. et al. (2011) 'RAIphy: phylogenetic classification of metagenomics samples using iterative refinement of relative abundance index profiles.', BMC bioinformatics. England, 12, p. 41. doi: 10.1186/1471-2105-12-41.

Namiki, T. et al. (2012) 'MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads.', Nucleic acids research. England, 40(20), p. e155. doi:

10.1093/nar/gks678.
National Center for Biotechnology Information - Nucleotide (1988). Available at: https://www.ncbi.nlm.nih.gov/nucleotide/ (Accessed: 2 August 2017).
National Center for Biotechnology Information - Pubmed (1946). Available at: https://www.ncbi.nlm.nih.gov/pubmed (Accessed: 2 August 2017).
Nawrocki, E. P. and Eddy, S. R. (2013) 'Infernal 1.1: 100-fold faster RNA homology searches.', Bioinformatics (Oxford, England). England, 29(22), pp. 2933–2935. doi: 10.1093/bioinformatics/btt509.
Nawrocki, E. P. et al. (2015) 'Rfam 12.0: updates to the RNA families database.', Nucleic acids research. England, 43(Database issue), pp. D130-7. doi: 10.1093/nar/gku1063.
Nelson, K. E. et al. (2010) 'A catalog of reference genomes from the human microbiome.', Science (New York, N.Y.). United States, 328(5981), pp. 994–999. doi: 10.1126/science.1183605.
Nguyen, N.-P. et al. (2014) 'TIPP: taxonomic identification and phylogenetic profiling.', Bioinformatics (Oxford, England). England, 30(24), pp. 3548–3555. doi: 10.1093/bioinformatics/btu721.
Niu, B. et al. (2011) 'FR-HIT, a very fast program to recruit metagenomic reads to homologous reference genomes.', Bioinformatics (Oxford, England). England, 27(12), pp. 1704–1705. doi: 10.1093/bioinformatics/btr252.
Noguchi, H., Park, J. and Takagi, T. (2006) 'MetaGene: prokaryotic gene finding from environmental genome shotgun sequences.', Nucleic acids research. England, 34(19), pp. 5623–5630. doi: 10.1093/nar/gkl723.
O'Leary, N. A. et al. (2016) 'Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation.', Nucleic acids research. England, 44(D1), pp. D733-45. doi: 10.1093/nar/gkv1189.
Olsen, G. J., Larsen, N. and Woese, C. R. (1991) 'The ribosomal RNA database project.', Nucleic acids research. England, 19 Suppl, pp. 2017–2021.
Ounit, R. et al. (2015) 'CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers.', BMC genomics. England, 16, p. 236. doi: 10.1186/s12864-015-1419-2.
Overbeek, R. et al. (2005) 'The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes.', Nucleic acids research. England, 33(17), pp. 5691–5702. doi: 10.1093/nar/gki866.
Pagès, H., Aboyoun, P., Gentleman, R., DebRoy, S. (2017) Biostrings: String objects representing biological sequences, and matching algorithms. R package version 2.44.2. Available at: http://bioconductor.org/packages/release/bioc/html/Biostrings.html (Accessed: 2 August 2017).
Parks, D. H. et al. (2014) 'STAMP: statistical analysis of taxonomic and functional profiles.', Bioinformatics (Oxford, England). England, 30(21), pp. 3123–3124. doi: 10.1093/bioinformatics/btu494.
Parks, D. H., MacDonald, N. J. and Beiko, R. G. (2011) 'Classifying short genomic fragments from novel lineages using composition and homology.', BMC bioinformatics. England, 12, p. 328. doi: 10.1186/1471-2105-12-328.
Peng, Y. et al. (2011) 'Meta-IDBA: a de Novo assembler for metagenomic data.', Bioinformatics (Oxford, England). England, 27(13), pp. i94-101. doi: 10.1093/bioinformatics/btr216.
Peng, Y. et al. (2012) 'IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth.', Bioinformatics (Oxford, England). England, 28(11), pp. 1420–1428. doi: 10.1093/bioinformatics/bts174.
Prabhakara, S. and Acharya, R. (2012) 'Unsupervised two-way clustering of metagenomic sequences.', Journal of biomedicine & biotechnology. United States, 2012, p. 153647. doi:

10.1155/2012/153647.
Prestat, E. et al. (2014) ‘FOAM (Functional Ontology Assignments for Metagenomes): a Hidden Markov Model (HMM) database with environmental focus.’, Nucleic acids research. England, 42(19), p. e145. doi: 10.1093/nar/gku702.
Pruesse, E., Peplies, J. and Glockner, F. O. (2012) ‘SINA: accurate high-throughput multiple sequence alignment of ribosomal RNA genes.’, Bioinformatics (Oxford, England). England, 28(14), pp. 1823–1829. doi: 10.1093/bioinformatics/bts252.
Pruitt, K. D. et al. (2012) ‘NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy.’, Nucleic acids research. England, 40(Database issue), pp. D130-5. doi: 10.1093/nar/gkr1079.
Puente-Sanchez, F., Aguirre, J. and Parro, V. (2016) ‘A novel conceptual approach to read-filtering in high-throughput amplicon sequencing studies.’, Nucleic acids research. England, 44(4), p. e40. doi: 10.1093/nar/gkv1113.
Quast, C. et al. (2013) ‘The SILVA ribosomal RNA gene database project: improved data processing and web-based tools.’, Nucleic acids research. England, 41(Database issue), pp. D590-6. doi: 10.1093/nar/gks1219.
Quince, C. et al. (2009) ‘Accurate determination of microbial diversity from 454 pyrosequencing data.’, Nature methods. United States, 6(9), pp. 639–641. doi: 10.1038/nmeth.1361.
Quince, C. et al. (2011) ‘Removing noise from pyrosequenced amplicons.’, BMC bioinformatics. England, 12, p. 38. doi: 10.1186/1471-2105-12-38.
Rawat, A. et al. (2014) ‘MetaGeniE: characterizing human clinical samples using deep metagenomic sequencing.’, PloS one. United States, 9(11), p. e110915. doi: 10.1371/journal.pone.0110915.
Reddy, R. M., Mohammed, M. H. and Mande, S. S. (2012) ‘TWARIT: an extremely rapid and efficient approach for phylogenetic classification of metagenomic sequences.’, Gene. Netherlands, 505(2), pp. 259–265. doi: 10.1016/j.gene.2012.06.014.
Relman, D. A. (2015) ‘The Human Microbiome and the Future Practice of Medicine.’, JAMA. United States, 314(11), pp. 1127–1128. doi: 10.1001/jama.2015.10700.
Rho, M., Tang, H. and Ye, Y. (2010) ‘FragGeneScan: predicting genes in short and error-prone reads.’, Nucleic acids research. England, 38(20), p. e191. doi: 10.1093/nar/gkq747.
RNAcentral: The non-coding RNA sequence database (2015). Available at: http://rnacentral.org/ (Accessed: 29 July 2017).
Rosen, G. et al. (2008) ‘Metagenome fragment classification using N-mer frequency profiles.’, Advances in bioinformatics. Egypt, 2008, p. 205969. doi: 10.1155/2008/205969.
Rusch, D. B. et al. (2007) ‘The Sorcerer II Global Ocean Sampling expedition: northwest Atlantic through eastern tropical Pacific.’, PLoS biology. United States, 5(3), p. e77. doi: 10.1371/journal.pbio.0050077.
Sandberg, R. et al. (2001) ‘Capturing whole-genome characteristics in short sequences using a naive Bayesian classifier.’, Genome research. United States, 11(8), pp. 1404–1409. doi: 10.1101/gr.186401.
Sanger, F. et al. (1977) ‘Nucleotide sequence of bacteriophage phi X174 DNA.’, Nature. England, 265(5596), pp. 687–695.
Sanli, K. et al. (2013) ‘FANTOM: Functional and taxonomic analysis of metagenomes.’, BMC bioinformatics. England, 14, p. 38. doi: 10.1186/1471-2105-14-38.
Scheuch, M., Hoper, D. and Beer, M. (2015) ‘RIEMS: a software pipeline for sensitive and comprehensive taxonomic classification of reads from metagenomics datasets.’, BMC bioinformatics. England, 16, p. 69. doi: 10.1186/s12859-015-0503-6.
Schloss, P. D. and Handelsman, J. (2005) ‘Introducing DOTUR, a computer program for defining operational taxonomic units and estimating species richness.’, Applied and environmental

microbiology. United States, 71(3), pp. 1501–1506. doi: 10.1128/AEM.71.3.1501-1506.2005.
Schloss, P. D. and Handelsman, J. (2006) 'Introducing SONS, a tool for operational taxonomic unit-based comparisons of microbial community memberships and structures.', Applied and environmental microbiology. United States, 72(10), pp. 6773–6779. doi: 10.1128/AEM.00474-06.
Schloss, P. D. and Handelsman, J. (2006) 'Introducing TreeClimber, a test to compare microbial community structures.', Applied and environmental microbiology. United States, 72(4), pp. 2379–2384. doi: 10.1128/AEM.72.4.2379-2384.2006.
Schloss, P. D. et al. (2009) 'Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities.', Applied and environmental microbiology. United States, 75(23), pp. 7537–7541. doi: 10.1128/AEM.01541-09.
Schloss, P. D., Larget, B. R. and Handelsman, J. (2004) 'Integration of microbial ecology and statistics: a test to compare gene libraries.', Applied and environmental microbiology. United States, 70(9), pp. 5485–5492. doi: 10.1128/AEM.70.9.5485-5492.2004.
Schmieder, R. and Edwards, R. (2011a) 'Fast identification and removal of sequence contamination from genomic and metagenomic datasets.', PloS one. United States, 6(3), p. e17288. doi: 10.1371/journal.pone.0017288.
Schmieder, R. and Edwards, R. (2011b) 'Quality control and preprocessing of metagenomic datasets.', Bioinformatics (Oxford, England). England, 27(6), pp. 863–864. doi: 10.1093/bioinformatics/btr026.
Scholz, M., Lo, C.-C. and Chain, P. S. G. (2014) 'Improved assemblies using a source-agnostic pipeline for MetaGenomic Assembly by Merging (MeGAMerge) of contigs.', Scientific reports. England, 4, p. 6480. doi: 10.1038/srep06480.
Schreiber, F. et al. (2010) 'Treephyler: fast taxonomic profiling of metagenomes.', Bioinformatics (Oxford, England). England, 26(7), pp. 960–961. doi: 10.1093/bioinformatics/btq070.
Schubert, M., Lindgreen, S. and Orlando, L. (2016) 'AdapterRemoval v2: rapid adapter trimming, identification, and read merging.', BMC research notes. England, 9, p. 88. doi: 10.1186/s13104-016-1900-2.
Segata, N. et al. (2011) 'Metagenomic biomarker discovery and explanation.', Genome biology. England, 12(6), p. R60. doi: 10.1186/gb-2011-12-6-r60.
Segata, N. et al. (2012) 'Metagenomic microbial community profiling using unique clade-specific marker genes.', Nature methods. United States, 9(8), pp. 811–814. doi: 10.1038/nmeth.2066.
Shamsaddini, A. et al. (2014) 'Census-based rapid and accurate metagenome taxonomic profiling.', BMC genomics, 15(1), p. 918. doi: 10.1186/1471-2164-15-918.
Shrestha, R. K. et al. (2014) 'QTrim: a novel tool for the quality trimming of sequence reads generated using the Roche/454 sequencing platform.', BMC bioinformatics. England, 15, p. 33. doi: 10.1186/1471-2105-15-33.
Simpson, J. T. and Durbin, R. (2012) 'Efficient de novo assembly of large genomes using compressed data structures.', Genome research. United States, 22(3), pp. 549–556. doi: 10.1101/gr.126953.111.
Singleton, D. R. et al. (2001) 'Quantitative comparisons of 16S rRNA gene sequence libraries from environmental samples.', Applied and environmental microbiology. United States, 67(9), pp. 4374–4376.
Sogin, M. L. et al. (2006) 'Microbial diversity in the deep sea and the underexplored "rare biosphere".', Proceedings of the National Academy of Sciences of the United States of America. United States, 103(32), pp. 12115–12120. doi: 10.1073/pnas.0605127103.
Stoddard, S. F. et al. (2015) 'rrnDB: improved tools for interpreting rRNA gene abundance in bacteria and archaea and a new foundation for future development.', Nucleic acids research. England, 43(Database issue), pp. D593-8. doi: 10.1093/nar/gku1201.
Sturm, M., Schroeder, C. and Bauer, P. (2016) 'SeqPurge: highly-sensitive adapter trimming for

paired-end NGS data.', BMC bioinformatics. England, 17, p. 208. doi: 10.1186/s12859-016-1069-7.

Su, X. et al. (2014a) 'Parallel-META 2.0: enhanced metagenomic data analysis with functional annotation, high performance computing and advanced visualization.', PloS one. United States, 9(3), p. e89323. doi: 10.1371/journal.pone.0089323.

Su, X. et al. (2014b) 'GPU-Meta-Storms: computing the structure similarities among massive amount of microbial community samples using GPU.', Bioinformatics (Oxford, England). England, 30(7), pp. 1031–1033. doi: 10.1093/bioinformatics/btt736.

Su, X., Xu, J. and Ning, K. (2012a) 'Parallel-META: efficient metagenomic data analysis based on high-performance computation.', BMC systems biology. England, 6 Suppl 1, p. S16. doi: 10.1186/1752-0509-6-S1-S16.

Su, X., Xu, J. and Ning, K. (2012b) 'Meta-Storms: efficient search for similar microbial communities based on a novel indexing scheme and similarity score for metagenomic data.', Bioinformatics (Oxford, England). England, 28(19), pp. 2493–2501. doi: 10.1093/bioinformatics/bts470.

Sunagawa, S. et al. (2013) 'Metagenomic species profiling using universal phylogenetic marker genes.', Nature methods. United States, 10(12), pp. 1196–1199. doi: 10.1038/nmeth.2693.

Tanaseichuk, O., Borneman, J. and Jiang, T. (2014) 'Phylogeny-based classification of microbial communities.', Bioinformatics (Oxford, England). England, 30(4), pp. 449–456. doi: 10.1093/bioinformatics/btt700.

Tatusov, R. L. et al. (2000) 'The COG database: a tool for genome-scale analysis of protein functions and evolution.', Nucleic acids research. England, 28(1), pp. 33–36.

Tatusova, T. et al. (2016) 'NCBI prokaryotic genome annotation pipeline.', Nucleic acids research. England, 44(14), pp. 6614–6624. doi: 10.1093/nar/gkw569.

Thompson, J. D., Higgins, D. G. and Gibson, T. J. (1994) 'CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.', Nucleic acids research. England, 22(22), pp. 4673–4680.

Treangen, T. J. et al. (2013) 'MetAMOS: a modular and open source metagenomic assembly and analysis pipeline.', Genome biology. England, 14(1), p. R2. doi: 10.1186/gb-2013-14-1-r2.

Triman, K. L. (1994) 'The 16S ribosomal RNA mutation database (16SMDB)', Nucleic acids research. England, 22(17), pp. 3563–3565.

Triman, K. L. (1996) 'The 16S ribosomal RNA mutation database (16SMDB).', Nucleic acids research. England, 24(1), pp. 166–168.

Triman, K. L. (2007) 'Mutational analysis of the ribosome.', Advances in genetics. United States, 58, pp. 89–119. doi: 10.1016/S0065-2660(06)58004-6.

Triman, K. L. and Adams, B. J. (1997) 'Expansion of the 16S and 23S ribosomal RNA mutation databases (16SMDB and 23SMDB).', Nucleic acids research. England, 25(1), pp. 188–191.

Triman, K. L., Peister, A. and Goel, R. A. (1998) 'Expanded versions of the 16S and 23S ribosomal RNA mutation databases (16SMDBexp and 23SMDBexp)', Nucleic acids research. England, 26(1), pp. 280–284.

Truong, D. T. et al. (2015) 'MetaPhlAn2 for enhanced metagenomic taxonomic profiling.', Nature methods. United States, pp. 902–903. doi: 10.1038/nmeth.3589.

Tuzhikov, A., Panchin, A. and Shestopalov, V. I. (2014) 'TUIT, a BLAST-based tool for taxonomic classification of nucleotide sequences.', BioTechniques. England, 56(2), pp. 78–84. doi: 10.2144/000114135.

Tyson, G. W. et al. (2004) 'Community structure and metabolism through reconstruction of microbial genomes from the environment.', Nature. England, 428(6978), pp. 37–43. doi: 10.1038/nature02340.

Van de Peer, Y. et al. (2000) 'The European small subunit ribosomal RNA database.', Nucleic acids

research. England, 28(1), pp. 175–176.
Vervier, K. et al. (2016) ‘Large-scale machine learning for metagenomics sequence classification.’, Bioinformatics (Oxford, England). England, 32(7), pp. 1023–1032. doi: 10.1093/bioinformatics/btv683.
Wang, Q. et al. (2007) ‘Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy.’, Applied and environmental microbiology. United States, 73(16), pp. 5261–5267. doi: 10.1128/AEM.00062-07.
Wang, X. et al. (2013) ‘M-pick, a modularity-based method for OTU picking of 16S rRNA sequences.’, BMC bioinformatics. England, 14, p. 43. doi: 10.1186/1471-2105-14-43.
Wang, X. et al. (2015) ‘MetaBoot: a machine learning framework of taxonomical biomarker discovery for different microbial communities based on metagenomic data.’, PeerJ. United States, 3, p. e993. doi: 10.7717/peerj.993.
Wang, Y. et al. (2012) ‘MetaCluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample.’, Bioinformatics (Oxford, England). England, 28(18), pp. i356–i362. doi: 10.1093/bioinformatics/bts397.
Wang, Y. et al. (2014) ‘MetaCluster-TA: taxonomic annotation for metagenomic data based on assembly-assisted binning.’, BMC genomics. England, 15 Suppl 1, p. S12. doi: 10.1186/1471-2164-15-S1-S12.
Wattam, A. R. et al. (2014) ‘PATRIC, the bacterial bioinformatics database and analysis resource.’, Nucleic acids research. England, 42(Database issue), pp. D581-91. doi: 10.1093/nar/gkt1099.
Web of Science (2014). Available at: https://apps.webofknowledge.com/ (Accessed: 11 August 2017).
Westcott, S. L. and Schloss, P. D. (2017) ‘OptiClust, an Improved Method for Assigning Amplicon-Based Sequence Data to Operational Taxonomic Units.’, mSphere. United States, 2(2). doi: 10.1128/mSphereDirect.00073-17.
White, J. R., Nagarajan, N. and Pop, M. (2009) ‘Statistical methods for detecting differentially abundant features in clinical metagenomic samples.’, PLoS computational biology. United States, 5(4), p. e1000352. doi: 10.1371/journal.pcbi.1000352.
Woese, C. R. et al. (1975) ‘Conservation of primary structure in 16S ribosomal RNA.’, Nature. England, 254(5495), pp. 83–86.
Wood, D. E. and Salzberg, S. L. (2014) ‘Kraken: ultrafast metagenomic sequence classification using exact alignments.’, Genome biology. England, 15(3), p. R46. doi: 10.1186/gb-2014-15-3-r46.
Wright, E. S., Yilmaz, L. S. and Noguera, D. R. (2012) ‘DECIPHER, a search-based approach to chimera identification for 16S rRNA sequences.’, Applied and environmental microbiology. United States, 78(3), pp. 717–725. doi: 10.1128/AEM.06516-11.
Wuyts, J., Perriere, G. and Van De Peer, Y. (2004) ‘The European ribosomal RNA database.’, Nucleic acids research. England, 32(Database issue), pp. D101-3. doi: 10.1093/nar/gkh065.
Xia, L. C. et al. (2011) ‘Accurate genome relative abundance estimation based on shotgun metagenomic reads.’, PloS one. United States, 6(12), p. e27992. doi: 10.1371/journal.pone.0027992.
Yang, Y., Zhong, C. and Yooseph, S. (2015) ‘SFA-SPA: a suffix array based short peptide assembler for metagenomic data.’, Bioinformatics (Oxford, England). England, 31(11), pp. 1833–1835. doi: 10.1093/bioinformatics/btv052.
Yano, M. et al. (2014) ‘CLAST: CUDA implemented large-scale alignment search tool.’, BMC bioinformatics. England, 15, p. 406. doi: 10.1186/s12859-014-0406-y.
Ye, Y., Choi, J.-H. and Tang, H. (2011) ‘RAPSearch: a fast protein similarity search tool for short reads.’, BMC bioinformatics. England, 12, p. 159. doi: 10.1186/1471-2105-12-159.

Yilmaz, P. et al. (2014) ‘The SILVA and “All-species Living Tree Project (LTP)” taxonomic frameworks.’, Nucleic acids research. England, 42(Database issue), pp. D643-8. doi: 10.1093/nar/gkt1209.

Yoon, S.-H. et al. (2017) ‘Introducing EzBioCloud: a taxonomically united database of 16S rRNA gene sequences and whole-genome assemblies.’, International journal of systematic and evolutionary microbiology. England, 67(5), pp. 1613–1617. doi: 10.1099/ijsem.0.001755.

Yu, F. et al. (2009) ‘GSTaxClassifier: a genomic signature based taxonomic classifier for metagenomic data analysis.’, Bioinformation. Singapore, 4(1), pp. 46–49.

Zaharia, M., Bolosky, W.J., Curtis, K., Fox, A., Patterson, D., Shenker, S., Stoica, I., Karp, R.M., Sittler, T. (2011) ‘Faster and More Accurate Sequence Alignment with SNAP’, eprint arXiv:1111.5572, pp. 1–10. Available at: https://arxiv.org/abs/1111.5572v1.

Zerbino, D. R. and Birney, E. (2008) ‘Velvet: algorithms for de novo short read assembly using de Bruijn graphs.’, Genome research. United States, 18(5), pp. 821–829. doi: 10.1101/gr.074492.107.

Zhang, Y., Sun, Y. and Cole, J. R. (2014) ‘A scalable and accurate targeted gene assembly tool (SAT-Assembler) for next-generation sequencing data.’, PLoS computational biology. United States, 10(8), p. e1003737. doi: 10.1371/journal.pcbi.1003737.

Zhang, Z. et al. (2000) ‘A greedy algorithm for aligning DNA sequences.’, Journal of computational biology : a journal of computational molecular cell biology. United States, 7(1–2), pp. 203–214. doi: 10.1089/10665270050081478.

Zhao, Y., Tang, H. and Ye, Y. (2012) ‘RAPSearch2: a fast and memory-efficient protein similarity search tool for next-generation sequencing data.’, Bioinformatics (Oxford, England). England, 28(1), pp. 125–126. doi: 10.1093/bioinformatics/btr595.

Zhou, Q. et al. (2013) ‘QC-Chain: fast and holistic quality control method for next-generation sequencing data.’, PloS one. United States, 8(4), p. e60234. doi: 10.1371/journal.pone.0060234.

Zhou, Q. et al. (2014) ‘Meta-QC-Chain: comprehensive and fast quality control method for metagenomic data.’, Genomics, proteomics & bioinformatics. China, 12(1), pp. 52–56. doi: 10.1016/j.gpb.2014.01.002.

Anexos

Anexo A. *Script Matlab* para criação de gráfico da distribuição de *Poisson*

```
distrP1=makedist('Poisson', 'lambda', 100);%cria a distribuição de Poisson com lambda=100
distrP2=makedist('Poisson', 'lambda', 500);%cria a distribuição de Poisson com lambda=500
distrP3=makedist('Poisson', 'lambda', 1000);%cria a distribuição de Poisson com lambda=1000
x=1:1:1200;%cria o vector x composto pelos números inteiros entre 1 e 1200
pdf1=pdf(distrP1, x);%calcula a função de densidade de probabilidade para a distribuição distrP1 com
os valores de x
pdf2=pdf(distrP2, x);%calcula a função de densidade de probabilidade para a distribuição distrP2 com
os valores de x
pdf3=pdf(distrP3, x);%calcula a função de densidade de probabilidade para a distribuição distrP3 com
os valores de x
figure;
plot(pdf1,'b', 'LineWidth', 1.5);%cria o gráfico para a função pdf1
hold on
plot(pdf2,'r', 'LineWidth', 1.5);%cria o gráfico para a função pdf2
plot(pdf3,'k', 'LineWidth', 1.5);%cria o gráfico para a função pdf3
legend({'lambda=100', 'lambda=500', 'lambda=1000'}, 'Location', 'NE');
title('distribuição de Poisson');
xlabel('número de amplicões');
ylabel('densidade de probabilidade');
hold off;
```

Anexo B. *Script Matlab* para criação de gráficos da distribuição *Half-Normal*

```
distrHN1=makedist('HalfNormal', 'mu', 1, 'sigma', 50);%cria a distribuição HalfNormal com mu=1 e sigma=50
distrTrunc1=truncate(distrHN1, 1, 150);%trunca a distribuição distrHN1 entre 1 e 150
distrHN2=makedist('HalfNormal', 'mu', 1, 'sigma', 80);%cria a distribuição HalfNormal com mu=1 e sigma=80
distrTrunc2=truncate(distrHN2, 1, 250);%trunca a distribuição distrHN2 entre 1 e 250
x=1:1:300;%cria o vector x composto pelos números inteiros entre 1 e 300
pdf1=pdf(distrHN1, x);%calcula a função de densidade de probabilidade para a distribuição distrHN1 com os valores de x
cdf1=cdf(distrTrunc1, x);%calcula a função de distribuição cumulativa para a distribuição distrTrunc1 com os valores de x
pdf2=pdf(distrHN2, x);%calcula a função de densidade de probabilidade para a distribuição distrHN2 com os valores de x
cdf2=cdf(distrTrunc2, x);%calcula a função de distribuição cumulativa para a distribuição distrTrunc2 com os valores de x
pdf3=pdf(distrTrunc1, x);%calcula a função de densidade de probabilidade para a distribuição distrtrunc1 com os valores de x
pdf4=pdf(distrTrunc2, x);%calcula a função de densidade de probabilidade para a distribuição distrTrunc2 com os valores de x
figure;
plot(pdf1,'b', 'LineWidth', 1.5);%cria o gráfico para a função pdf1
hold on
plot(pdf2,'r', 'LineWidth', 1.5);%cria o gráfico para a função pdf2
legend({'mu=1, sigma=50', 'mu=1, sigma=80'}, 'Location', 'NE');
title('Distribuição Half-Normal');
xlabel('comprimento da leitura');
ylabel('densidade de probabilidade');
hold off;
figure;
plot(pdf3,'b', 'LineWidth', 1.5);%cria o gráfico para a função pdf3
hold on
plot(pdf4,'r', 'LineWidth', 1.5);%cria o gráfico para a função pdf4
legend({'mu=1, sigma=50', 'mu=1, sigma=80'}, 'Location', 'NE');
title('Distribuição Half-Normal truncada');
xlabel('comprimento da leitura');
ylabel('densidade de probabilidade');
hold off;
figure;
plot(cdf1,'b', 'LineWidth', 1.5);%cria o gráfico para a função cdf1
hold on
plot(cdf2,'r', 'LineWidth', 1.5);%cria o gráfico para a função cdf2
legend({'mu=1, sigma=50', 'mu=1, sigma=80'}, 'Location', 'SE');
title('Distribuição Half-Normal truncada');
xlabel('comprimento da leitura');
```

```
ylabel('probabilidade cumulativa');
hold off;
```

Anexo C. Funções *Matlab* do programa *sim16S*

```
function sim16S(fastaFile, numSeq, primerF, primerR, readLen, mutReads, numMut)
%A função sim16S gera 10000 leituras simuladas de sequenciação do gene 16S rRNA. Esta função
%usa 8 funções sequenciais para gerar o conjunto de dados simulados. As sequências do gene 16S
%rRNA são seleccionadas com base em sequências de oligonucleótidos introduzidas pelo utilizador a
%partir de uma base de dados de sequências de referência. O utilizador pode definir o comprimento
%das leituras de sequenciação, o número de leituras com erros (substituições de bases) e o número de
%erros por leitura.

%O argumento fastaFile é o nome do ficheiro das sequências de referência (em formato fasta).
%O argumento numSeq corresponde ao número total de sequências a extrair da base de dados de
%sequências de referência.
%Os argumentos primerF e primerR correspondem às sequências dos oligonucleótidos forward e
%reverse (introduzidas no sentido 5'-3').
%O argumento readLen é o comprimento máximo das leituras de sequenciação.
%O argumento mutReads estabelece a probabilidade de se obter uma leitura com erros (quanto maior
%o valor de mutReads, menor será a probabilidade de se obter uma leitura com erros).
%O argumento numMut é o número de erros de cada leitura.

tstart=tic;%inicia a contagem do tempo de execução do programa
[totalSeq] = seqConc(fastaFile, numSeq);%executa a função seqConc
[maxLength, minLength, avLength, ampCount] = seqAmplicon(primerF, primerR, numSeq);%executa
%a função seqAmplicon
[speciesSet, ampliconSet, uniqueSeq, ampCount] = compareAmp(ampCount);%executa a função
%compareAmp
[nSpecies] = randAmp(ampCount, speciesSet, ampliconSet);%executa a função randAmp
countTaxa;%executa a função countTaxa
[readLen] = readLength(readLen);%executa a função readLength
[totalMutReads] = mutAmp(readLen, mutReads, numMut);%executa a função mutAmp
telapsed=toc(tstart);%termina a contagem do tempo de execução do programa
sim16Sreport(fastaFile, primerF, primerR, numSeq, readLen, mutReads, numMut, totalSeq,
ampCount, maxLength, minLength, avLength, uniqueSeq, totalMutReads, nSpecies,
telapsed);%executa a função sim16Sreport
fclose('all');

end


function [totalSeq] = seqConc(fastaFile, numSeq)
%A função seqConc faz a concatenação das linhas de sequência para cada entrada existente no
%ficheiro das sequências de referência.

%O argumento fastaFile é o nome do ficheiro das sequências de referência (em formato fasta).
%O argumento numSeq corresponde ao número total de sequências a concatenar.
%A função gera os ficheiros 'refSeq.fasta' (sequências concatenadas) e 'taxonomyRef.txt' (taxonomia)
%para cada entrada da base de dados e devolve o número total de bases das sequências concatenadas
```

```
%(totalSeq).

inFile1=fopen(fastaFile, 'r');
outFile1=fopen('refSeq.fasta', 'w+');
outFile2=fopen('taxonomyRef.txt', 'w+');
header=fgets(inFile1);%obtém a primeira linha do ficheiro fasta
totalSeq=0;%inicializa a contagem do número total de bases das sequências concatenadas

for i=1:numSeq
   fprintf(outFile1, header);%escreve o identificador da sequência no ficheiro 'refSeq.fasta'
   fprintf(outFile2, num2str(header(2:end)));%escreve o identificador da sequência (sem >) no ficheiro
%'taxonomyRef.txt'
   line=fgets(inFile1);%obtém a 1ª linha da sequência do gene 16S rRNA
   sequence='';

   while line(1)~='>';%adiciona cada nova linha de sequência à linha anterior

      for j=1:numel(line)%converte as bases U para T da sequência original

         if line(j)=='U'
            line(j)='T';
         end

      end

      sequence=strcat(sequence, line);
      line=fgets(inFile1);
      header=line;%sai do ciclo quando a linha começa com '>'

   end

   totalSeq=totalSeq+numel(sequence);%calcula o total de bases das sequências concatenadas
   fprintf(outFile1, sequence);%escreve a sequência concatenada do gene 16S rRNA no ficheiro
%'refSeq.fasta'
   fprintf(outFile1, '\r\n');

end

end


function [maxLength, minLength, avLength, ampCount] = seqAmplicon (primerF, primerR, numSeq)
%A função seqAmplicon gera sequências de amplicões do gene 16S rRNA a partir de um par de
%oligonucleótidos específicos. Nota: A função só pode ser aplicada se existir, no máximo, uma única
%sequência idêntica a cada oligonucleótido em cada sequência de referência.

%Os argumentos primerF e primerR correspondem às sequências dos oligonucleótidos forward e
%reverse (introduzidas no sentido 5'-3').
```

```
%O argumento numSeq corresponde ao número de sequências extraídas da base de dados de
%sequências de referência.
%A função devolve o número de amplicões seleccionados (ampCount) e os comprimentos máximo
%(maxLength), mínimo (minLength) e médio (avLength) dos amplicões.
%A função gera os ficheiros 'ampliconSet.txt' (sequências dos amplicões seleccionados),
'speciesSet.txt' %(identificadores das sequências seleccionadas) e 'excludedSeq.fasta' (identificadores e
sequências %excluídas).

maxLength=0;
minLength=10000;
lengthSum=0;
ampCount=0;
inFile2=fopen('refSeq.fasta', 'r');
outFile3=fopen('ampliconSet.txt', 'w+');
outFile4=fopen('speciesSet.txt', 'w+');
outFile5=fopen('excludedSeq.fasta', 'w+');
primerRC=primerR;%iguala a sequência do primerR à  do primerRC (reverse+complement) para fazer
%a pré-alocação

for i=0:numel(primerR)-1%escreve a sequência complementar invertida do primer reverse

   if primerR(end-i)=='A'
      primerRC(1+i)='T';
   elseif primerR(end-i)=='C'
      primerRC(1+i)='G';
   elseif primerR(end-i)=='G'
      primerRC(1+i)='C';
   elseif primerR(end-i)=='T'
      primerRC(1+i)='A';
   end

end

for i=1:numSeq
   header=fgets(inFile2);
   sequence=fgets(inFile2);
   startPos=strfind(sequence, primerF);%devolve a posição da 1ª base do lado 5' do primer forward se
%existente
   endPos=strfind(sequence, primerRC)+numel(primerRC)-1;%devolve a posição da 1ª base do lado 5'
%do primer reverse se existente

   if endPos>startPos
      amplicon=sequence(startPos:endPos);
      fprintf(outFile4, header);
      fprintf(outFile3, amplicon);
      fprintf(outFile3, '\r\n');

      if numel(amplicon)>maxLength%determina o comprimento máximo do amplicão
```

```
        maxLength=numel(amplicon);
    end

    if numel(amplicon)<minLength%determina o comprimento mínimo do amplicão
        minLength=numel(amplicon);
    end

  lengthSum=lengthSum+numel(amplicon);%determina o somatório dos comprimentos dos
%amplicões
  ampCount=ampCount+1;%determina o total de amplicões seleccionados

  else
    fprintf(outFile5, header);
    fprintf(outFile5, sequence);
  end

end

avLength=lengthSum/ampCount;%calcula a média do comprimento dos amplicões

end


function [speciesSet, ampliconSet, uniqueSeq, ampCount] = compareAmp(ampCount)
%A função compareAmp compara as sequências dos amplicões entre si para determinar o número de
%sequências iguais.

%O argumento ampCount é o número de amplicões seleccionados com a função seqAmplicon.
%A função devolve o número de sequências distintas (uniqueSeq) e o conjunto de espécies e de
%amplicões seleccionados sob a forma de cell arrays.
%A função gera o ficheiro 'repeatSeq.txt' (número de vezes que cada amplicão aparece repetido no
%conjunto seleccionado).

inFile3=fopen('ampliconSet.txt', 'r');
inFile4=fopen('speciesSet.txt', 'r');
outFile6=fopen('repeatSeq.txt', 'w+');
ampliconSet=textscan(inFile3, '%s');%cria um cell array com 1 sequência por célula
speciesSet=textscan(inFile4, '%s', 'Delimiter', '\r');%cria um cell array com um
%identificador/taxonomia por célula
uniqueSeq=0;

for i=1:ampCount

  duplicateSeq=-1;%a contagem começa em -1 para descontar a comparação de cada sequência
%consigo própria

  for j=1:ampCount
```

```
        if strcmp(ampliconSet{1}{i}, ampliconSet{1}{j})==1
            duplicateSeq=duplicateSeq+1;
        end

    end

    if duplicateSeq==0
        uniqueSeq=uniqueSeq+1;
    end

    fprintf(outFile6, speciesSet{1}{i});%escreve o identificador e taxonomia de cada sequência
    fprintf(outFile6, '\r');
    fprintf(outFile6, num2str(duplicateSeq));%escreve o número de sequências iguais abaixo do
%identificador/taxonomia de cada sequência
    fprintf(outFile6, '\r');

end

end

function [nSpecies] = randAmp(ampCount, speciesSet, ampliconSet)
%A função randAmp gera 10000 sequências aleatoriamente de acordo com uma distribuição de
%Poisson prévia aplicada ao número de amplicões. Esta distribuição pretende simular a estrutura de
%uma comunidade de microorganismos, em que a abundância dos diferentes táxones pode variar
bastante %entre si.

%O argumento ampCount é o número de amplicões seleccionados com a função seqAmplicon.
%Os argumentos speciesSet e ampliconSet são os cell arrays produzidos com a função compareAmp.
%A função devolve o número total de espécies (nSpecies) obtido após randomização dos amplicões.
%A função gera os ficheiros 'countSpecies.txt' (número de vezes que cada sequência está presente no
%conjunto de 10000 sequências) e 'ampliconRand.txt' (sequências dos 10000 amplicões
seleccionados).

outFile7=fopen('countSpecies.txt', 'w+');
outFile8=fopen('ampliconRand.txt', 'w+');
distrP=makedist('Poisson', 'lambda', ampCount/2);%cria uma distribuição de Poisson com média e
%variância igual a ampCount/2
rng('shuffle');%modifica o gerador de números aleatórios de acordo com a hora de execução do
%programa
randMatrix=random(distrP, 100, 100);%gera uma matriz de 10000 números inteiros aleatórios de
%acordo com a distribuição de Poisson
nSpecies=0;

    for i=1:ampCount
        logicMatrix=randMatrix==i;%gera uma matriz lógica de presença/ausência de cada amplicão na
%matriz randMatrix
        s=sum(logicMatrix(1:end));%soma todas as ocorrências do amplicão na matriz
```

```
        if s>0 %são seleccionados apenas os valores que aparecem pelo menos 1 vez na matriz

            fprintf(outFile7, speciesSet{1}{i});%escreve o identificador/taxonomia de cada sequência
            fprintf(outFile7, '\r');
            fprintf(outFile7, num2str(s));%escreve o número de ocorrências do amplicão
            fprintf(outFile7, '\r');
            nSpecies=nSpecies+1;

            for j=1:s
            fprintf(outFile8, ampliconSet{1}{i});%escreve a sequência do amplicão
            fprintf(outFile8, '\r');
            end

        end

    end

end


function countTaxa
%A função countTaxa conta o número de sequências dos vários táxones em cada nível taxonómico.

%A função gera um ficheiro ('taxaStatistics.txt') que contém o número de sequências atribuídas a cada
%táxon, organizadas por nível taxonómico.

inFile5=fopen('countSpecies.txt', 'r');
taxaSet=textscan(inFile5, '%s %s %s %s %s %s %s', 'Delimiter', ';');%cria um
%cell array que contém 7 colunas, em que a primeira corresponde ao
%identificador da sequência e as restantes 6 aos diferentes níveis
%taxonómicos (ordenados do domínio para a espécie)
outFile9=fopen('taxaStatistics.txt', 'w+');
level = {'domain', 'phylum', 'class', 'order', 'family', 'genus', 'species'};
fprintf(outFile9, '<Taxa statistics>');
fprintf(outFile9, '\r');

    for i=2:7 %percorre as 6 últimas colunas do cell array

        fprintf(outFile9, '\r');
        fprintf(outFile9, strcat(num2str(level{i}),':'));%adiciona a designação do nível taxonómico
        fprintf(outFile9, '\r');
        list=categorical(taxaSet{i});%lista todos os táxones de cada nível taxonómico
        taxaList=categories(list);%lista cada táxon de cada nível taxonómico uma única vez
        total=0;

        for j=1:length(taxaList)%percorre todas as categorias de táxones

            abundance=0;
```

```
        for k=1:length(taxaSet{i})%percorre todas as linhas do cell array

            if strcmp(taxaList(j), num2str(taxaSet{i}{k}))==1%compara cada categoria com cada táxon
                abundance=abundance+str2double(taxaSet{1}{k+1});%adiciona o número de sequências
%de cada táxon à respectiva categoria
            end

        end

        fprintf(outFile9, char(taxaList(j)));
        fprintf(outFile9, '_');
        fprintf(outFile9, num2str(abundance));
        fprintf(outFile9, '\r');
        total=total+abundance;%calcula o total de sequências de cada nível taxonómico

    end

  fprintf(outFile9, 'Total sequences:');
  fprintf(outFile9, num2str(total));
  fprintf(outFile9, '\r');
  fprintf(outFile9, 'Number of different taxa:');
  fprintf(outFile9, num2str(numel(taxaList)));
  fprintf(outFile9, '\r');

  end

end


function [readLen] = readLength (readLen)
%A função readLength trunca o tamanho das leituras ao valor de readLen. Nota 1: O argumento
%readLen só pode tomar os valores 150 e 250. Nota 2: A função só pode ser aplicada se o tamanho do
%amplicão for igual ou superior ao valor de readLen.

%A função gera um ficheiro ('readSet.txt') que contém as leituras truncadas.

inFile6=fopen('ampliconRand.txt', 'r');
outFile10=fopen('readSet.txt', 'w+');

  for i=1:10000
    amplicon=fgets(inFile6);
    fprintf(outFile10, amplicon(1:readLen));
    fprintf(outFile10, '\r');
  end

end
```

```
function [totalMutReads] = mutAmp (readLen, mutReads, numMut)
%A função mutAmp introduz substituições de bases nas sequências geradas pela função readLength.
%Cada base é substituída por uma das restantes 3 bases. Nota 1: As posições de base mista (ambíguas)
%não são alteradas pela função. Nota 2: No caso de se pretender criar 2 ou mais substituições por
%sequência, as substituições após a primeira poderão ocorrer numa posição anteriormente alterada.

%O argumento readLen é o comprimento máximo das sequências.
%O argumento mutReads estabelece o limite superior do intervalo com base no qual são seleccionadas
%aleatoriamente as sequências com substituições.
%O argumento numMut é o número de substituições por sequência.
%A função devolve o número de sequências que contêm substituições (totalMutReads).
%A função gera o ficheiro 'mutSet.fasta' que contém o conjunto total de leituras após introdução das
%substituições, em que cada leitura está identificada sequencialmente de seq1 a seq10000.

inFile7=fopen('readSet.txt', 'r');
outFile11=fopen('mutSet.fasta', 'w+');

if readLen==150
   sig=50;%o valor de sigma para a distribuição HalfNormal foi escolhido de forma a permitir
%introduzir substituições em todo o comprimento da leitura de 150 bases
elseif readLen==250
   sig=80;%o valor de sigma para a distribuição HalfNormal foi escolhido
%de forma a permitir introduzir substituiçõesem todo o comprimento da leitura de 250 bases
end

distrHN=makedist('HalfNormal', 'mu', 1, 'sigma', sig);%cria uma distribuição HalfNormal com média
%igual a 1 e sigma igual a sig, para usar na seleção da posição a substituir
distrTrunc=truncate(distrHN, 1, readLen);%trunca a distribuição HalfNormal entre as posições 1 e
%readLen
seq1='CGT'; seq2='AGT'; seq3='ACT'; seq4='ACG';
totalMutReads=0;
rng('shuffle');%modifica o gerador de números aleatórios de acordo com a hora de execução do
%programa

   for i=1:10000
      read=fgets(inFile7);
      fprintf(outFile11, '>seq');
      fprintf(outFile11, num2str(i));
      fprintf(outFile11, '\n');
      selection=randi([1 mutReads]);%o intervalo para seleção aleatória das leituras depende do valor
%de mutReads. Quanto maior for o valor de mutReads, menor será a probabilidade de se obter uma
%leitura com substituições (selection=1).

      if selection==1
         totalMutReads=totalMutReads+1;

         for j=1:numMut
```

```
            position=round(random(distrTrunc));%a posição da substituição é escolhida aleatoriamente
%com base na distribuição HalfNormal truncada
            mutPosition=readLen-position+1;%a posição de substituição é corrigida de forma a que as
%substituições ocorram mais próximo do final das leituras
            m=randi(3);

            if read(mutPosition)=='A'
               read(mutPosition)=seq1(m);
            elseif read(mutPosition)=='C'
               read(mutPosition)=seq2(m);
            elseif read(mutPosition)=='G'
               read(mutPosition)=seq3(m);
            elseif read(mutPosition)=='T'
               read(mutPosition)=seq4(m);
            end

         end

      mutAmplicon=read;
      fprintf(outFile11, num2str(mutAmplicon));
      fprintf(outFile11, '\n');

      else

      fprintf(outFile11, num2str(read));
      fprintf(outFile11, '\n');

      end

   end

end


function sim16Sreport (fastaFile, primerF, primerR, numSeq, readLen, mutReads, numMut, totalSeq,
ampCount, maxLength, minLength, avLength, uniqueSeq, totalMutReads, nSpecies, telapsed)
%A função sim16Sreport produz um relatório de execução do programa sim16S.

%A função gera o ficheiro 'report.txt' que contém os parâmetros de entrada e os dados de saída do
%programa.

outFile12=fopen('report.txt', 'w+');

fprintf(outFile12, '< sim16S Report >');
fprintf(outFile12, '\r');
fprintf(outFile12, '\r');
fprintf(outFile12, 'total run time (in seconds): ');
fprintf(outFile12, num2str(telapsed));
```

```
fprintf(outFile12, '\r');
fprintf(outFile12, '\r');
fprintf(outFile12, 'Input data:');
fprintf(outFile12, '\r');
fprintf(outFile12, '\r');
fprintf(outFile12, ' reference 16S rRNA database: ');
fprintf(outFile12, fastaFile);
fprintf(outFile12, '\r');
fprintf(outFile12, ' forward primer sequence: ');
fprintf(outFile12, primerF);
fprintf(outFile12, '\r');
fprintf(outFile12, ' reverse primer sequence: ');
fprintf(outFile12, primerR);
fprintf(outFile12, '\r');
fprintf(outFile12, ' number of sequences screened: ');
fprintf(outFile12, num2str(numSeq));
fprintf(outFile12, '\r');
fprintf(outFile12, ' maximum read length: ');
fprintf(outFile12, num2str(readLen));
fprintf(outFile12, '\r');
fprintf(outFile12, ' number of predicted mutated reads: ');
fprintf(outFile12, num2str((1/mutReads)*10000));
fprintf(outFile12, '\r');
fprintf(outFile12, ' number of mutations per read: ');
fprintf(outFile12, num2str(numMut));
fprintf(outFile12, '\r');
fprintf(outFile12, '\r');
fprintf(outFile12, 'Output data:');
fprintf(outFile12, '\r');
fprintf(outFile12, '\r');
fprintf(outFile12, ' total number of bases processed: ');
fprintf(outFile12, num2str(totalSeq));
fprintf(outFile12, '\r');
fprintf(outFile12, ' number of amplicons selected: ');
fprintf(outFile12, num2str(ampCount));
fprintf(outFile12, '\r');
fprintf(outFile12, ' percentage of primers on target: ');
fprintf(outFile12, num2str(ampCount/numSeq*100));
fprintf(outFile12, '\r');
fprintf(outFile12, ' maximum amplicon length: ');
fprintf(outFile12, num2str(maxLength));
fprintf(outFile12, '\r');
fprintf(outFile12, ' minimum amplicon length: ');
fprintf(outFile12, num2str(minLength));
fprintf(outFile12, '\r');
fprintf(outFile12, ' average amplicon length: ');
fprintf(outFile12, num2str(round(avLength)));
fprintf(outFile12, '\r');
```

```
fprintf(outFile12, ' number of unique amplicon sequences: ');
fprintf(outFile12, num2str(uniqueSeq));
fprintf(outFile12, '\r');
fprintf(outFile12, ' percentage of redundant sequences: ');
fprintf(outFile12, num2str((ampCount-uniqueSeq)/ampCount*100));
fprintf(outFile12, '\r');
fprintf(outFile12, ' number of taxa in final dataset: ');
fprintf(outFile12, num2str(nSpecies));
fprintf(outFile12, '\r');
fprintf(outFile12, ' number of mutated reads: ');
fprintf(outFile12, num2str(totalMutReads));
fprintf(outFile12, '\r');
fprintf(outFile12, ' overall mutation percentage: ');
fprintf(outFile12, num2str((totalMutReads*numMut/(10000*readLen))*100));
fprintf(outFile12, '\r');

end
```

Anexo D. Cálculo do total de sequências em cada táxon por nível taxonómico

1) Abrir o ficheiro de texto correspondente a um dado nível taxonómico (por exemplo, *otu_table_mc2_w_tax_no_pynast_failures_L2.txt*) no *Microsoft Excel*;

2) Substituir os pontos (.) por vírgulas (,) em toda a folha de cálculo;

3) Introduzir uma nova coluna entre as colunas A e B;

4) Na célula B3 (livre) correspondente à linha do primeiro táxon, introduzir uma fórmula para calcular a soma dos valores de todas as colunas da linha 3 situadas à direita dessa célula;

5) Copiar a fórmula para as restantes linhas da tabela de táxones;

6) Introduzir uma fórmula na coluna B, após a última linha da tabela, para calcular o total de sequências de todas as linhas da coluna B;

7) Gravar o ficheiro no formato 'Livro do Excel'.

Anexo E. Configuração do ficheiro *taxonomyRef.txt*

1) Abrir ficheiro *taxonomyRef.txt* produzido pelo *sim16S* no programa *LibreOffice Calc*;

2) Nas opções de separadores, seleccionar 'Espaço'; pressionar 'Aceitar';

3) Inserir uma nova coluna entre as colunas A e B;

4) Na primeira linha da nova coluna escrever a seguinte fórmula: =C1&D1&E1&F1&G1&H1&I1&J1&K1&L1&";";

5) Copiar fórmula para as restantes linhas da tabela;

6) Copiar a coluna B para a coluna M, colando apenas o texto e não as fórmulas;

7) Remover colunas B a L;

8) Gravar o ficheiro *taxonomyRef* com terminação *.csv*;

9) Abrir o ficheiro *taxonomyRef.csv* no editor de texto *gedit* para confirmar que a formatação está correcta;

10) Fechar o ficheiro e alterar manualmente a extensão *.csv* para *.txt*;

11) Abrir o ficheiro *taxonomyRef.csv* no programa *Microsoft Excel*;

12) Substituir espaço em branco por vírgula; confirmar que o total de substituições é igual ao número de sequências (linhas) do ficheiro;

13) Seleccionar a coluna A, o tabulador 'Dados' e de seguida 'Texto para colunas';

14) Seleccionar 'Delimitado', Delimitador=vírgula e Formato dos dados da coluna=texto;

15) Confirmar que a primeira coluna tem o identificador taxonómico e a segunda coluna tem a descrição taxonómica completa;

16) Procurar espaços em branco nas colunas do ficheiro e confirmar que não existem;

17) Guardar ficheiro no formato 'texto (separado por tabulações)';

18) Abrir ficheiro no editor de texto *gedit*;

19) Substituir aspas (") por espaço vazio;

20) Gravar o ficheiro.

Anexo F. Criação de ficheiro de configuração de bases de dados personalizadas para uso no *QIIME*

Por forma a que o *QIIME* possa usar outras bases de dados de sequências de referência e respectiva taxonomia, em vez das bases de dados incluídas por defeito, é necessário criar um ficheiro em formato *txt* designado *.qiime.config*, que é colocado em */home/luis*, e no qual são especificados os ficheiros que contêm as sequências de referência e a respectiva taxonomia. Este ficheiro deverá conter em cada linha o nome do parâmetro configurável e o respetivo ficheiro associado (incluindo localização), separados por uma tabulação, de acordo com o seguinte exemplo:

assign_taxonomy_reference_seqs_fp */home/luis/taxonomyRef.txt*
pick_otus_reference_seqs_fp */home/luis/refSeq.fasta*
assign_taxonomy_id_to_taxonomy_fp */home/luis/taxonomyRef.txt*
assign_taxonomy_reference_seqs_fp */home/luis/refSeq.fasta*

Anexo G. Comandos de execução do programa *QIIME*

O *QIIME* foi executado usando os seguintes comandos:

1) Activar o ambiente do programa:

$ *source activate qiime1*

2) Proceder à identificação de UTO(s):

$ *pick_open_reference_otus.py -i /home/luis/mutSet.fasta -o /home/luis/qiime_results --minimum_failure_threshold 10 -s 1 -a*

3) Sumarizar os dados do ficheiro por nível taxonómico:

$ *summarize_taxa.py -i /home/luis/qiime_results/otu_table_mc2_w_tax_no_pynast_failures.biom -o /home/luis/qiime_results/taxa --suppress_biom_table_output -a*

4) Desactivar o ambiente do programa:

$ *source deactivate qiime1*

Anexo H. Comandos de execução do programa *mothur*

As 6 etapas seguintes representam um exemplo de comandos do *mothur* para análise de leituras do amplicão A usando a base de dados de sequências *SILVA* personalizada (1000 sequências):

1) Proceder à remoção de bases ambíguas e de sequências duplicadas, construir a colecção de sequências de referência e alinhar as leituras contra esta colecção.

$ *./mothur "#summary.seqs(fasta=/home/luis/Programas/mothur/mutSet.fasta);screen.seqs(fasta=/home/luis/Programas/mothur/mutSet.fasta,group=/home/luis/Programas/mothur/mutSet.groups,maxambig=0);summary.seqs(fasta=/home/luis/Programas/mothur/mutSet.good.fasta);unique.seqs(fasta=/home/luis/Programas/mothur/mutSet.good.fasta);count.seqs(name=/home/luis/Programas/mothur/mutSet.good.names,group=/home/luis/Programas/mothur/mutSet.good.groups);pcr.seqs(fasta=rep_set_aligned.fasta,oligos=pcrTestAmplA.oligos,processors=4);align.seqs(fasta=/home/luis/Programas/mothur/mutSet.good.unique.fasta,reference=rep_set_aligned.scrap.pcr.fasta);summary.seqs(fasta=/home/luis/Programas/mothur/mutSet.good.unique.align,count=/home/luis/Programas/mothur/mutSet.good.count_table)"*

Nota: Na execução do comando *align.seqs* é utilizada como sequência de referência o ficheiro *rep_set_aligned.scrap.pcr.fasta* e não o ficheiro *rep_set_aligned.pcr.fasta* conforme indicado no tutorial. Esta alteração deveu-se ao facto de o ficheiro *rep_set_aligned.pcr.fasta* se encontrar vazio após a corrida do comando *pcr.seqs*, enquanto o ficheiro *rep_set_aligned.scrap.pcr.fasta* continha todas as sequência alinhadas. O facto de o ficheiro *rep_set_aligned.pcr.fasta* se encontrar vazio pode estar relacionado com a ausência da sequência do oligonucleótido *reverse* (que é indicada através do ficheiro de oligos incluído no comando *pcr.seqs*) nas sequências alinhadas, uma vez que o comprimento fixo das leituras (150 ou 250 pb) eliminou a região do oligonucleótido *reverse*. Apesar de esta ser apenas uma possibilidade, todos os conjuntos de dados foram analisados usando o ficheiro *rep_set_aligned.scrap.pcr.fasta* e, em função dos resultados obtidos, esta utilização não pareceu afectar de forma visível a classificação taxonómica.

2) Efectuar o agrupamento de sequências e a classificação das mesmas com o classificador Bayesiano implementado no programa *RDP Classifier*. As posições do alinhamento *StartPosition* e *endPosition* devem ser definidas em função do resultado do comando *summary.seqs* anterior.

$ *./mothur "#screen.seqs(fasta=/home/luis/Programas/mothur/mutSet.good.unique.align,count=/home/luis/Programas/mothur/mutSet.good.count_table,summary=/home/luis/Programas/mothur/mutSet.good.unique.summary,start=startPosition,end=endPosition);summary.seqs(fasta=current,count=current);filter.seqs(fasta=/home/luis/Programas/mothur/mutSet.good.unique.good.align,vertical=T,trump=.);unique.seqs(fasta=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.fasta,count=/home/luis/Programas/mothur/mutSet.good.good.count_table);pre.cluster(fasta=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.fasta,count=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.count_table,diffs=1);classify.seqs(fasta=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.fasta,count=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.count_table,reference=refSeq.fasta,taxonomy=taxonomyRef.txt,method=wang,processors=4,output=simple)"*

3) Proceder à classificação em UTO(s), derivar a taxonomia consenso e criar um ficheiro *.biom*.

$ *./mothur "#dist.seqs(fasta=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.fasta,cutoff=0.20);cluster(column=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.dist,count=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.count_table);make.shared(list=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.an.unique_list.list,count=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.count_table);classify.otu(list=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.an.unique_list.list,count=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.count_table,taxonomy=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.taxonomyRef.wang.taxonomy);make.biom(shared=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.an.unique_list.shared,constaxonomy=/home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.an.unique_list.unique.cons.taxonomy)"*

4) Activar o ambiente do programa *QIIME*:

$ *source activate qiime1*

5) Sumarizar os dados do ficheiro por nível taxonómico:

$ *summarize_taxa.py –i /home/luis/Programas/mothur/mutSet.good.unique.good.filter.unique.precluster.an.unique_list.unique.biom -o /home/luis/Programas/mothur/taxa --suppress_biom_table_output –a*

6) Desactivar o ambiente para o programa *qiime*:

$ *source deactivate qiime1*

Anexo I. Criação de ficheiro de grupos para utilização no programa *mothur*

1) Abrir um programa editor de texto (e.g., *gedit*);

2) No início da primeira linha escrever o identificador da sequência da leitura (e.g., seq1);

3) No início da segunda linha escrever a designação do grupo a que pertence a sequência (e.g., A);

4) Repetir os passos 2) e 3) para o total de sequências existentes no ficheiro mutSet.fasta;

5) Guardar o documento como ficheiro de texto, atribuindo a extensão *groups* (e.g., mutSet.groups);

6) Colocar o ficheiro na pasta do programa *mothur*.

Anexo J. Criação de ficheiros de sequências de oligonucleótidos para utilização no programa *mothur*

1) Abrir um programa editor de texto (e.g., *gedit*);

2) No início da primeira linha escrever 'primer' e inserir uma tabulação;

3) Após a tabulação, escrever a sequência do oligonucleótido *forward* (letras maiúsculas) no sentido 5'-3' e inserir uma tabulação;

4) Após a tabulação, escrever a sequência do oligonucleótido *reverse* (letras maiúsculas) no sentido 5'-3' e inserir uma tabulação;

5) Após a tabulação, escrever a designação da região hipervariável que corresponde à localização dos oligonucleótidos (e.g., V3);

6) Guardar o documento como ficheiro de texto, atribuindo a extensão *oligos* (e.g., pcrTestAmplA.oligos);

7) Colocar o ficheiro na pasta do programa *mothur*.

Anexo L. Variação do número de leituras do *sim16S* processadas pelo *mothur*

#	Conjunto de dados do *sim16S* (a)	Leituras sem bases ambíguas	Leituras únicas (pré-alinhamento)	Leituras únicas (pós-alinhamento)	Leituras classificadas (pós-agrupamento)
1	A_1k_150_noMutations	9672	77	76	72
2	B_1k_250_noMutations	8919	77	77	73
3	A_1k_150_1%_1_rep1	9672	190	160	77
4	A_1k_150_1%_1_rep2	9672	194	149	76
5	A_1k_150_10%_1_rep1	9672	1057	813	104
6	A_1k_150_10%_1_rep2	9672	968	718	103
7	A_1k_150_100%_1_rep1	9672	6357	4880	421
8	A_1k_150_100%_1_rep2	9672	6362	4866	526
9	A_1k_150_1%_1	9658	164	147	73
10	A_1k_150_10%_1	9662	978	718	103
11	A_1k_150_100%_1	9681	6396	4922	466
12	B_1k_250_1%_1	8990	151	150	75
13	B_1k_250_10%_1	8933	978	920	78
14	B_1k_250_100%_1	9029	6951	6644	75
15	A_1k_150_10%_2	9672	1069	927	631
16	A_1k_150_10%_4	9672	1082	1065	954
17	B_5k_250_noMutations	9879	181	180	143
18	A_1k_150_1%_1	9658	164	148	76
19	A_1k_150_10%_1	9662	978	758	105
20	A_1k_150_100%_1	9681	6396	5188	1569

(a) Cada conjunto de dados do *sim16* é constituído por 10000 leituras.